

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-230249

(43)Date of publication of application : 10.09.1996

(51)Int.CI.

B41J 5/30

B41J 2/485

G06F 9/45

G06T 1/00

(21)Application number : 07-040935

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 28.02.1995

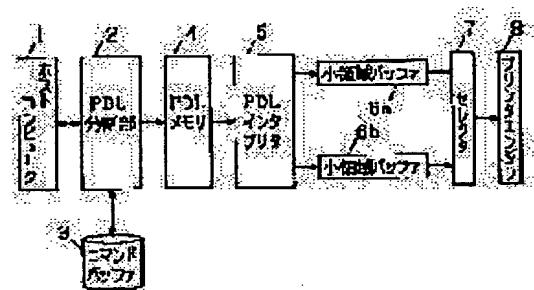
(72)Inventor : TSUTSUMI KENJI

(54) IMAGE PROCESSING APPARATUS

(57)Abstract:

PURPOSE: To provide an image processing apparatus capable of shortening an image developing time when an image is developed at every small regions being divided image regions in response to PDL command data.

CONSTITUTION: PDL command data is transmitted to a PDL dividing part 2 from a host computer 1 to be temporarily stored in a command buffer 3. An image region is divided into a plurality of small regions. A part of the PDL command data is divided into PDL command data at every small regions to be stored in a PDL memory 4. When there is a printing start command, a PDL interrupter 5 is started and the PDL command data read from the PDL memory 4 are developed as image data at every small regions to be alternately written in small region buffers 6a, 6b. At each time when writing is completed, a printing request is issued to a printer engine 8. This operation is repeated until the printing of all of small regions is completed.



LEGAL STATUS

[Date of request for examination] 14.12.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3336800

[Date of registration] 09.08.2002

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-230249

(43)公開日 平成8年(1996)9月10日

(51)Int.Cl.*	識別記号	庁内整理番号	F I	技術表示箇所
B 41 J 5/30			B 41 J 5/30	Z
2/485			3/12	P
G 06 F 9/45		7737-5B	G 06 F 9/44	3 2 0 C
G 06 T 1/00			15/62	K

審査請求 未請求 請求項の数3 O.L (全37頁)

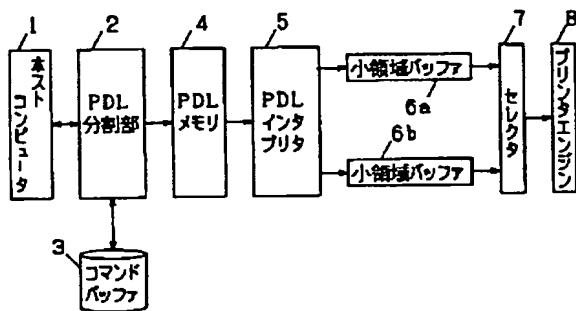
(21)出願番号	特願平7-40935	(71)出願人	000005496 富士ゼロックス株式会社 東京都港区赤坂二丁目17番22号
(22)出願日	平成7年(1995)2月28日	(72)発明者	堤 健次 神奈川県海老名市本郷2274番地 富士ゼロ ックス株式会社内
		(74)代理人	弁理士 石井 康夫 (外1名)

(54)【発明の名称】 画像処理装置

(57)【要約】

【目的】 PDLコマンドデータに応じて、分割された画像領域である小領域ごとに画像展開する際に、画像展開時間を短縮できる画像処理装置を提供する。

【構成】 PDLコマンドデータは、ホストコンピュータ1からPDL分割部2に転送され、コマンドバッファ3に一時記憶される。画像領域は複数の小領域に分割される。PDLコマンドデータは、その一部のコマンドデータについては、小領域ごとのPDLコマンドデータに分割され、PDLメモリ4に記憶される。印刷開始コマンドがあると、PDLインタプリタ5が起動し、PDLメモリ4から読み出されたPDLコマンドデータは、小領域ごとの画像データに展開し、小領域バッファ6a, 6bに交互に書き込まれる。書き込みが終了する度に、プリンタエンジン8に対しプリント要求を行なう。全小領域のプリントが終了するまで、このような動作を繰り返す。



【特許請求の範囲】

【請求項1】 ページ記述言語で記載されたコマンドデータに応じて画像データを出力する画像処理装置において、前記コマンドデータを入力し一部のコマンドデータについて分割された画像領域である小領域ごとのコマンドデータに分割する分割手段と、該分割手段のコマンドデータ出力を記憶する記憶手段と、該記憶手段から読み出されたコマンドデータに応じて前記小領域ごとの画像データを出力するインタプリタを有することを特徴とする画像処理装置。

【請求項2】 前記分割手段は、入力されたコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるコマンドデータを各小領域ごとのコマンドデータとするものであることを特徴とする請求項1に記載の画像処理装置。

【請求項3】 前記分割手段は、入力されたコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるコマンドデータから各小領域の範囲内でのみ画像展開されるコマンドデータを作成し、作成されたコマンドデータを各小領域ごとのコマンドデータとするものであることを特徴とする請求項1に記載の画像処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ページ記述言語をサポートする画像処理装置に関するものである。

【0002】

【従来の技術】 コンピュータに接続された記録装置等のための画像処理装置において、ページ記述言語（以下、「PDL」という。）をサポートするものが、従来より知られている。図32は、ページ記述言語をサポートする画像処理装置の第1の従来例を示すブロック図である。図中、21はホストコンピュータ、22はPDLインタプリタ、23はコマンドバッファ、24は1ページ分の画像メモリ、25はプリンタエンジン、26はインターフェースである。このプリンタエンジン25は、レザープリンタなどにおけるように、1ページ分の画像メモリから画像データの供給を受ける必要があるタイプのものである。

【0003】 ホストコンピュータ21は、PDLコマンドデータを出力する。PDLインタプリタ22は、インターフェース26を介して、PDLコマンドデータを取り込み、コマンドバッファ23に記憶した上で、1ページ分のPDLコマンドデータに基づいて画像データを1ページ分の画像メモリ24上に展開する。PDLインタプリタ22は、この展開を終了すると、終了をプリンタエンジン25に知らせ、プリンタエンジン25は、1ページ分の画像メモリ26に記憶された画像データを読み出

10

20

30

40

50

して1ページ分の印字を行なう。プリンタエンジン25は、印字終了後、この終了をPDLインタプリタ22に知らせ、PDLインタプリタ22は、次のページのPDLコマンドデータに対し、再び同様の処理をする。

【0004】 図33は、ページ記述言語をサポートする画像処理装置の第2の従来例を示すブロック図である。この第2の従来例は、例えば、特開平5-31974号公報等により知られている。図中、図32と同様な部分には同じ符号を用い説明を省略する。27は小領域バッファ、28は圧縮器、29は画像圧縮メモリ、30は復号器、31、32は信号線である。この従来例においては、1ページ分の画像領域が複数の小領域である、領域0ないし領域7に分割されている。小領域バッファ27は、1つの小領域の画像データの容量に相当する容量のバッファである。

【0005】 PDLインタプリタ22は、領域0から順に各小領域ごとに、ページ記述言語で記載されたコマンドデータ（以下、「PDLコマンドデータ」という。）の1ページ分に基づいて、当該小領域内の画像データのみを小領域バッファ27上に展開する。

【0006】 圧縮器28は、PDLインタプリタ22の指示を信号線31を介して受け、小領域バッファ27に記憶された当該小領域内の画像データを圧縮符号化により圧縮画像データに変換し、順次1ページ分の画像圧縮メモリ29に記憶させる。圧縮器28は、1小領域分の画像データの圧縮符号化を終了すると、信号線31を介しPDLインタプリタ22に終了を知らせる。そうすると、PDLインタプリタ22および圧縮器28は、次の小領域内の画像データに対し、再び、PDLコマンドデータの1ページ分に基づいて同様の処理をし、最後の領域7までこの処理を繰り返す。

【0007】 このようにして、1ページ分の画像データの展開とその圧縮符号化とを終了すると、信号線32を介して復号器30を起動する。復号器30は、画像圧縮メモリ29に記憶された圧縮符号化データを読み出して復号し、プリンタエンジン25に出力する。このようにして、プリンタエンジン25は1ページ分の印字を行なう。復号器30は、復号終了後、終了を知らせる信号を信号線32を介してPDLインタプリタ22に出力する。

【0008】 図34は、ページ記述言語をサポートする画像処理装置の第3の従来例を示すブロック図である。この第3の従来例も、例えば、特開平5-31974号公報等により知られている。図中、図34と同様の部分には同じ符号を用い説明を省略する。27aは第1の小領域バッファ、27bは第2の小領域バッファ、33はマルチブレクサ、34はセレクタ、35、36は信号線である。この第3の従来例は、図33に示されるブロック図において、小領域バッファ27として、同容量の第1および第2の小領域バッファ27a、27bを使用し

た場合の例であり、そのために、マルチブレクサ33とセレクタ34を追加したものである。したがって、全体的な動作については、上述した図33に示されるブロック図と同様であり、相違する部分について動作を説明する。

【0009】マルチブレクサ33は、PDLインタプリタ22の指示を信号線35を介して受け、PDLインタプリタ22の出力を第1の小領域バッファ27aが第2の小領域バッファ27bのいずれかに交互に接続し、PDLインタプリタ22の出力をいずれかに書き込ませる。セレクタ34は、信号線36を介してPDLインタプリタ22の指示を受け、第1の小領域バッファ27aの書き込み期間中は、第2の小領域バッファ27bを読み出して圧縮器28に出力し、第2の小領域バッファ27bの書き込み期間中は、第1の小領域バッファ27aを読み出して圧縮器28に出力する。小領域バッファ27として2つの第1および第2の小領域バッファ27a, 27bを使用するから、PDLインタプリタ22による書き込み動作と圧縮器28による読み出し動作と同時に実行することができ、処理時間が短縮される。

【0010】1ページ単位で画像データ展開することを前提としたPDLは、第1の従来例のように1ページ分の画像メモリから画像データの供給を受ける必要があるタイプのプリンタエンジンに適している。

【0011】しかし、シリアルプリンタのプリンタエンジンのように1ページ分の画像データの連続供給を必要としないもののための画像処理装置においては、1ページ分の画像メモリを備えていない場合が多い。また、そのようなプリンタのために1ページ分の画像メモリを備えることはコストアップにつながる。そのため、1ページ単位で画像データ展開することを前提としたPDLをサポートするには、PDLコマンドデータをそのまま画像展開することができず、何らかの処理が必要となる。

【0012】また、1ページ分の画像メモリを有する画像処理装置においても、画像処理過程において、第2、第3の従来例の小領域バッファ24等のように、1ページの画像データ分の容量を持たない画像メモリにPDLコマンドデータを画像展開する必要がある場合には、何らかの処理が必要となる。

【0013】第2、第3の従来例においては、画像データの1ページ分に満たない容量の画像メモリである小領域バッファ24等にPDLコマンドデータを画像展開するに際し、PDLインタプリタ22は、領域0から順に各小領域ごとに、1ページ分のPDLコマンドデータに基づいて、当該小領域内の画像データのみを小領域バッファ24上に展開し、その後、次の小領域内の画像データに対し、再び同様の画像処理をしていた。したがって、ある1つの小領域の画像データを小領域バッファ24等の上に画像展開する際には、他の小領域にのみ画像データ展開されるようなコマンドデータも含めて、1ペ

ージ分全体のPDLコマンドデータを読み出して、この1つの小領域に画像データ展開されるコマンドデータか否かを判定しなければならないから、その分、画像展開時間が長くなり、画像の送出速度を大きくすることができない。さらに、1ページ分全てのPDLコマンドデータの処理を小領域の数だけ繰り返し行なうものから、小領域の数が多くなるほど、1ページ全体を処理する時間が長くなるという問題もあった。

【0014】

【発明が解決しようとする課題】本発明は、上述した事情に鑑みてなされたもので、PDLコマンドデータに応じて、分割された画像領域である小領域ごとに画像展開する際の画像展開時間を短縮することができ、画像の送出速度を大きくすることができる画像処理装置を提供することを目的とするものである。

【0015】

【課題を解決するための手段】本願の発明は、請求項1に記載の発明においては、ページ記述言語で記載されたコマンドデータに応じて画像データを出力する画像処理装置において、前記コマンドデータを入力し一部のコマンドデータについて分割された画像領域である小領域ごとのコマンドデータに分割する分割手段と、該分割手段のコマンドデータ出力を記憶する記憶手段と、該記憶手段から読み出されたコマンドデータに応じて前記小領域ごとの画像データを出力するインタプリタを有することを特徴とするものである。

【0016】請求項2に記載された発明においては、請求項1に記載の画像処理装置において、前記分割手段は、入力されたコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるコマンドデータを各小領域ごとのコマンドデータとするものであることを特徴とするものである。

【0017】請求項3に記載の発明においては、請求項1に記載の画像処理装置において、前記分割手段は、入力されたコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるコマンドデータから各小領域の範囲内でのみ画像展開されるコマンドデータを作成し、作成されたコマンドデータを各小領域ごとのコマンドデータとするものであることを特徴とするものである。

【0018】

【作用】請求項1に記載の発明によれば、コマンドデータを入力し一部のコマンドデータについて分割された画像領域である小領域ごとのコマンドデータに分割する分割手段と、この分割手段のコマンドデータ出力を記憶する記憶手段と、この記憶手段から読み出されたコマンドデータに応じて小領域ごとの画像データを出力するインタプリタを有するものであるから、インタプリタが1つ

の小領域の画像データを展開する際に、他の小領域にのみ画像データ展開されるようなコマンドデータを読み出しこの1つの小領域に画像データ展開されるコマンドデータか否かを判定する必要がないから、1つの小領域の画像展開時間を短くすることができ、画像の送出速度を大きくすることができる。また、例えば、展開された画像を保持するためのバッファが小さいプリンタ等にも適用することができる。

【0019】請求項2に記載の発明によれば、請求項1に記載の画像処理装置において、前記分割手段は、入力されたコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるコマンドデータを各小領域ごとのコマンドデータとするものであるから、請求項1に記載の発明と同様に、1つの小領域の画像展開時間を短くすることができ、画像の送出速度を大きくすることができ、また、例えば、展開された画像を保持するためのバッファが小さいプリンタ等にも適用することができる。さらに、小領域ごとのコマンドデータを得るために処理が簡単になる。

【0020】請求項3に記載の発明によれば、請求項1に記載の画像処理装置において、前記分割手段は、入力されたコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるコマンドデータから各小領域の範囲内でのみ画像展開されるコマンドデータを作成し、作成されたコマンドデータを各小領域ごとのコマンドデータとするものであるから、請求項1に記載の発明と同様に、1つの小領域の画像展開時間を短くすることができ、画像の送出速度を大きくすることができ、また、例えば、展開された画像を保持するためのバッファが小さいプリンタ等にも適用することができる。さらに、小領域ごとのコマンドデータを画像データに展開する際の処理が簡単になる。

【0021】

【実施例】図1は、本発明の一実施例のブロック図である。1はホストコンピュータ、2はPDL分割部、3はコマンドバッファ、4はPDLメモリ、5はPDLインタブリタ、6a, 6bは小領域バッファ、7はセレクタ、8はプリンタエンジンである。

【0022】PDLコマンドデータは、ホストコンピュータ1からPDL分割部2に転送される。PDL分割部2は、PDLコマンドデータをコマンドバッファ3に一時記憶させる。1ページ分の画像領域は複数の小領域に分割されている。PDL分割部2は、コマンドバッファ3に一時記憶されたPDLコマンドデータを入力し一部のコマンドデータについては、小領域ごとのPDLコマンドデータに分割し、PDLメモリ4に記憶させる。この作業をPDLコマンドデータ中に印刷開始コマンドが送られてくるまで繰り返す。

【0023】小領域ごとのPDLコマンドデータとしては、図7、8等を用いて後述する本発明の一実施例の第1の具体例においては、入力されたPDLコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータをそのままの形で各小領域ごとのPDLコマンドデータとする。その際、あるPDLコマンドが複数の小領域上に画像展開される場合には、この複数の小領域に同じPDLコマンドデータがそのままの形で割り当てられることになる。

【0024】また、図13、14等を用いて後述する本発明の一実施例の第2の具体例においては、入力されたPDLコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータから各小領域の範囲内でのみ画像展開されるPDLコマンドデータを作成し、作成されたPDLコマンドデータを各小領域ごとのPDLコマンドデータとする。その際、もとのPDLコマンドデータが複数の小領域上に画像展開されるものである場合には、この複数の小領域のそれぞれに対し、各小領域の範囲内でのみ画像展開されるPDLコマンドデータが作成される。

【0025】PDL分割部2においては、全てのPDLコマンドを小領域ごとのPDLコマンドデータに分割するのではなく、PDLコマンドの内容によっては、入力したPDLコマンドデータをそのまま出力する。複数の小領域にまたがって画像展開されるようなPDLコマンドデータについては、小領域ごとのPDLコマンドデータに分割する。しかし、一つの小領域にしか画像展開されないPDLコマンドデータや、各小領域に共通のPDLコマンドデータなど、その他のPDLコマンドデータについては、そのまま出力する。このような機能を、PDL分割部2の機能に含めたが、実際に装置を構成する際には、必ずしもPDL分割部2のブロック部に含める必要はなく、他のブロックを経由して、PDLメモリ4に書き込んでもよい。なお、複数の小領域にまたがって画像展開されるようなPDLコマンドデータ全てを、必ずしも小領域ごとのPDLコマンドデータに分割する必要はない。

【0026】上述したいづれの具体例においても、入力されたPDLコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出する際、2つの手法をとることができる。第1の手法は、1ページ分の最初のPDLコマンドデータから順に、PDLコマンドデータの一つを先に特定し、この特定されたPDLコマンドデータに応じた画像データの少なくとも一部が各小領域上に展開されるか否かを判定し、以後、次のPDLコマンドデータについて同様の判定をするというものである。第2の手法は、最初の領域から順に、小領域の一つを先に特定し、この特定された小領域上に1ページ分

のPDLコマンドデータに応じた画像データの少なくとも一部が展開されるか否かを判定し、以後、次の小領域について同様の判定をするというものである。

【0027】第1の手法によれば、同じ1ページ分のPDLコマンドデータの処理を小領域の数だけ繰り返し行なうものではないから、小領域の数が多くなっても、1ページ全体の処理時間がさほど長くならない。後述する第1、第2の具体例においては、第1の手法を用いている。

【0028】PDLメモリ4における記憶形態は任意であるが、例えば、画像展開される小領域ごとに区分してPDLコマンドデータを記憶させる。

【0029】PDLコマンドデータの中に印刷開始命令があると、PDLインタプリタ5が起動し、PDLメモリ4に記憶されたPDLコマンドデータがPDLインタプリタ5に渡される。

【0030】PDLインタプリタ5は、PDLコマンドデータを小領域バッファ6aまたは6bのいずれかに交互に展開して小領域ごとの画像データに展開する。

【0031】各小領域ごとのPDLコマンドデータが、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータのそのままの形である場合、PDLインタプリタ5は、画像データを展開しようとしている当該小領域の範囲のみを小領域バッファ6a、6bに展開する。しかし、当該小領域の範囲外の部分の画像データの作成は、これを行なわないか、行なったとしても小領域バッファ6a、6bへの書き込みを行なわない。

【0032】また、各小領域ごとのPDLコマンドデータが、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータを各小領域の範囲内でのみ画像展開されるPDLコマンドデータを作成したものである場合、小領域ごとのPDLコマンドデータは、あらかじめ小領域の範囲内でのみ画像展開されるようになっているから、PDLインタプリタ5は、各小領域ごとのPDLコマンドデータをそのまま画像データに展開すればよい。

【0033】小領域バッファ6aまたは6bのいずれかへの画像データの書き込みが終了する度に、PDLインタプリタ5は、プリンタエンジン8に対しプリント要求を行なう。セレクタ7は、第1の小領域バッファ6aが書き込み期間中のときは、第2の小領域バッファ6bを読み出し、第2の小領域バッファ6bが書き込み期間中のときは、第1の小領域バッファ6aを読み出してプリンタエンジン8に出力する。

【0034】すなわち、画像データの新たな書き込みが終了した方の小領域バッファは、セレクタ7により選択され読み出され、読み出された1小領域分の画像データは、プリンタエンジン8に転送される。PDLインタプリタ5は、プリント実行中も読み出されていない方の小領域バッファに対し、画像データを書き込むことがき

る。PDLインタプリタ5は、全小領域のプリントが終了するまでこのような動作を繰り返す。

【0035】なお、図34に示される第2の従来例のブロック図においては、小領域バッファ27a、27bがマルチブレクサ33により切り換えられていた。しかし、上述した実施例では、PDLインタプリタ5がマルチブレクサの機能も含むように構成しているので、マルチブレクサ33に相当するブロックを明示していない。また、ブロック間で制御信号をやりとりするための信号線の図示は省略されている。

【0036】小領域バッファは、6a、6bの2つとしたが、個数は任意である。小領域バッファの個数を3以上とし、書き込みを行なう小領域バッファを順次切り替え、書き込みの終了した小領域バッファを順次選択して読み出すようにしてもよい。逆に、プリンタエンジン8が、小領域バッファの書き込み中に小領域バッファを読み出せなくともよいタイプのもの、例えば、シリアルプリンタ用のものであれば、小領域バッファを1つにし、マルチブレクサの機能およびセレクタ7を除いてよい。

【0037】なお、1つの小領域の画像データが必要とするメモリ容量と、1つの小領域バッファのメモリ容量とは、必ずしも一致させる必要はない。例えば、小領域バッファのメモリ容量を2倍にし、2つの小領域分の画像データをまとめて1つの小領域バッファに書き込むようにもよい。また、採用するプリンタエンジン8が1ページ分の画像メモリから画像データの供給を受ける必要のあるタイプの場合には、セレクタの後段にこの1ページ分の画像メモリを設ける代わりに、PDLインタプリタ5の出力に直接1ページ分の画像メモリを設け、マルチブレクサの機能、および、小領域バッファ6a、6b、セレクタ7をなくしてもよい。

【0038】図2は、本発明の一実施例におけるハードウエア構成を説明する説明図である。図1に示した各ブロックは、このハードウエア構成により実現される。図1の機能的ブロックと直接的に対応する部分には同じ符号を用い説明を省略する。10は内部バス、11はCPU、12はROM、13はRAM、14はディスクコントローラ、15はディスク装置、16はホストインタフェース、17はバッファである。図中、破線で囲む部分は、図1におけるPDL分割部2からセレクタ7までのブロックに対応する。内部バス10上には、CPU11、ROM12、RAM13、ディスクコントローラ14、バッファ17、小領域バッファ6a、6bが接続される。そして、ディスクコントローラ14にはハードディスク15が接続され、バッファ17にはホストインタフェース16が接続され、小領域バッファ6a、6bにはセレクタ7が接続される。

【0039】ホストインタフェース16は、ホストコンピュータ1との間でデータを通信するための装置で、一

一般的には、例えば、イーサネットコントローラなどのネットワークインターフェースや、RS232Cコントローラ等のシリアルインターフェースやGPIB等のパラレルインターフェースが用いられる。バッファ17は、ホストコンピュータ1からホストインターフェース16を介して送られるPDLコマンドデータを一時記憶するものである。ホストインターフェース16は、このバッファ17が一杯になると、ホストコンピュータ1に対しPDLコマンドデータの伝送を一時停止するように要求する。

【0040】なお、このバッファ17の代わりにディスク装置15を用い、ホストコンピュータから伝送されてくるPDLコマンドデータが、全てディスク装置15に一時記憶されるようにしてもよい。その際、ホストインターフェース16から、直接に内部バス10、ディスクコントローラ14を経由して一時記憶されてもよいし、バッファ17を介してもよい。図1に示されるコマンドバッファ3は、バッファ17およびまたはディスク装置15により実現される。

【0041】CPU11は、ROM12に記憶されたプログラムにより、図1に示されるPDL分割部2の処理を実行する。図1に示されるPDLメモリ3の機能はRAM13により実現されるが、ディスク装置15により実現されてもよいし、両者により実現されてもよい。CPU11は、また、ROM12に記憶されたプログラムにより、図1に示されるPDLインターリタ5の処理も実行する。

【0042】図3から図6までを用いてホストコンピュータ1から送られるPDLコマンドデータの一例を説明する。図3は、画像領域の座標系の一例の説明図である。この一例において、画像領域は、左上の角を座標の原点とする。以後の記載では、位置の上下関係を説明するとき、この画像領域上でみた上下位置に基づいて説明する。したがって、副走査方向のY座標値が大きくなる方を下位置としている。小領域の境界線の左横に示した数字は、各境界の上と下に位置する画素のY座標値である。主走査方向にも複数の縦線を記載しているが、これは、単に座標位置を分かりやすくするための線である。この縦線の上に示した数字は、縦線の右に位置する最初の画素のX座標値である。

【0043】この例では、1ページの画像領域は、主走査方向であるX軸方向に6145画素、副走査方向であるY軸方向に8192画素からなる。いわゆる400dpiの画素密度においては、A3サイズは、X軸方向に4752画素、Y軸方向に6720画素の画像サイズとなり、A4サイズは、X軸方向に3360画素、Y軸方向に4752画素の画像サイズとなる。Y軸方向を1024画素ごとの8つの小領域に分割すると、1つの小領域は、 6144×1024 画素からなる。したがって、A3サイズは、領域0から領域6までの小領域からなり、A4サイズは、領域0から領域4までの小領域から

なる。領域を特定する番号を、以後、小領域ポインタという。

【0044】なお、一例として、小領域バッファ6a, 6bは、それぞれ、この 6144×1024 画素からなる小領域の各画素のデータを記憶し、1画素は、例えば、カラー画像の場合、一例としてRGB各8ビットからなる。

【0045】図4、図5は、PDLコマンドデータの一例を示す説明図である。この一例では、ページ記述言語として、Post Script（登録商標）を採用している。図4、図5における一連のPDLコマンドデータは、1ページ分の画像を出力するコマンドデータであり、ホストコンピュータ1から伝送され、コマンドバッファ3に書き込まれるものである。

【0046】図6は、図4、図5のPDLコマンドデータの実行結果を説明する説明図である。図中、Aは赤の第1の矩形、Bは緑の第2の矩形、Cは青の菱形、Dは黄の三角形であり、赤の第1の矩形Aは、一部を青の菱形Cにより上書きされ、緑の第2の矩形Bは、一部を黄の三角形Dで上書きされている。括弧で囲まれた2つの数字は、(0, 0)がA3サイズの用紙の原点のXY座標値を、その他が各図形A～Dの頂点の一部のXY座標値をPDLコマンドが使用する座標系で表わす。

【0047】図6も適宜参照しながら、図4、図5に示されるPDLコマンドデータの一例を説明する。図4の211～234は、定義に関するPDLコマンドが記載されている。211において、「pel」を $400 \div 72$ と定義する。212～219において、矩形のバスを定義する。213における「newpath」コマンドによりバスがクリアされる。214において、「move to」コマンドにより初期位置(0, 0)が設定される。なお、(0, 0)の最初の0はX座標が0であること、次の0はY座標が0であることを示す。以下、位置のX座標、Y座標を同様に表記する。215において、初期位置(0, 0)から「lineto」コマンドにより位置(0, 1)への第1番目のバスが設定される。216において、位置(0, 1)から「lineto」コマンドにより位置(1, 1)への第2番目のバスが設定される。217において、位置(1, 1)から「lineto」コマンドにより位置(1, 0)への第3番目のバスが設定される。218において、位置(1, 0)から「closepath」コマンドにより初期位置(0, 0)への第4番目のバスが設定される。

【0048】220～226において、三角形のバスを定義する。221における「newpath」コマンドによりバスがクリアされる。222において、「move to」コマンドにより初期位置(0, 0)が設定される。223において、初期位置(0, 0)から「lineto」コマンドにより位置(2, 0)への第1番目のバスが設定される。224において、位置(2, 0)か

11

ら「lineto」コマンドにより位置(1, 2)への第2番目のバスが設定される。224において、位置(1, 2)から「closepath」コマンドにより初期位置(0, 0)への第3番目のバスが設定される。

【0049】227～234において、菱形のバスを定義する。228における「newpath」コマンドによりバスがクリアされる。229において、「move to」コマンドにより初期位置(0, 0)が設定される。230において、初期位置(0, 0)から「lineto」コマンドにより位置(3, -2)への第1番目のバスが設定される。231において、位置(3, -2)から「lineto」コマンドにより位置(6, 0)への第2番目のバスが設定される。232において、位置(6, 0)から「lineto」コマンドにより位置(3, 2)への第3番目のバスが設定される。233において、位置(3, 2)から「closepath」コマンドにより初期位置(0, 0)への第4番目のバスが設定される。

【0050】図5の235において、「scale」コマンドにより、先に図4の211で定義された「pel」を用いて、スケーリングを行なう。X座標の1単位をpelの値に、Y座標の1単位をpelの値にスケーリングする。このスケーリングにより、CRT画面上の解像度75dpiの座標系をプリンタエンジン用の400dpiの座標系に変換する。236において、「gsave」コマンドにより現在の内部状態をセーブする。

【0051】237において、「translate」コマンドにより、初期位置を(512, 832)に設定し、238において、「scale」コマンドにより、X座標の1単位を768に、Y座標の1単位を2048にスケーリングする。239において、「square」により、描画図形として矩形が設定される。これは、先に定義されているから、図4の212～219においてなされた矩形の定義におけるバスを237における初期位置と、238におけるスケーリングにしたがって変換すると、実際に描画されるべき矩形のバスが設定される。

【0052】すなわち、図6における第1の矩形Aに示されるように、初期位置(512, 832)から位置(512, 1600)まで第1のバスが設定され、位置(512, 1600)から位置(2560, 1600)まで第2のバスが設定され、位置(2560, 1600)から位置(2560, 832)まで第3のバスが設定され、位置(2560, 832)から初期位置(512, 832)まで第4のバスが設定される。

【0053】240において、「setrgbcolor」コマンドにより、描画図形をペイントする色を赤に設定する。このコマンドの「100」の各数字は、それぞれ赤の成分、緑の成分、青の成分を指示し、240においては、赤を指示している。241において、「fi

10

20

30

40

50

12

11」コマンドにより、240において設定された色「赤」で図6における第1の矩形Aの内部が塗りつぶされる。

【0054】242において、「restore」コマンドにより、236においてセーブされた内部状態が復元される。243において、「translate」コマンドにより、初期位置を(1024, 3136)に設定し、244において、「scale」コマンドにより、X座標の1単位を2048に、Y座標の1単位を2048にスケーリングする。245において、「square」により、描画図形として矩形が設定される。これは、先に定義してあるから、図4の212～219においてなされた矩形の定義におけるバスを243における初期位置と、244におけるスケーリングにしたがって変換すると、実際に描画されるべき矩形のバスが設定される。

【0055】すなわち、図6における第2の矩形Bに示されるように、初期位置(1024, 3136)から位置(1024, 5184)まで第1のバスが設定され、位置(1024, 5184)から位置(3072, 5184)まで第2のバスが設定され、位置(3072, 5184)から位置(3072, 3136)まで第3のバスが設定され、位置(3072, 3136)から初期位置(1024, 3136)まで第4のバスが設定される。246において、「setrgbcolor」コマンドにより、描画図形をペイントする色を緑に設定する。247において、「fill」コマンドにより、246において設定された色「緑」で図6における第2の矩形B内が塗りつぶされる。

【0056】248において、「initgraphics」コマンドにより、内部状態を初期化する。249において、「scale」コマンドにより、図4の211で定義された「pel」を用いて、再度スケーリングを行なう。250において、「gsave」コマンドによりPDL分割部2の現在の内部状態をセーブする。

【0057】251において、「translate」コマンドにより、初期位置を(1024, 1088)に設定し、252において、「scale」コマンドにより、X座標の単位1を512に、Y座標の単位1を512にスケーリングする。253において、「diamond」により、描画図形として菱形が設定される。これは、先に定義してあるから、図4の227～234においてなされた菱形の定義におけるバスを251における初期位置と、252におけるスケーリングにしたがって変換すると、実際に描画されるべき菱形のバスが設定される。

【0058】すなわち、図6における菱形Cに示されるように、初期位置(1024, 1088)から位置(2560, 64)まで第1のバスが設定され、位置(2560, 64)から位置(4096, 1088)まで第2

のバスが設定され、位置(4096, 1088)から位置(2560, 2122)まで第3のバスが設定され、位置(2560, 2112)から初期位置(1024, 1088)まで第4のバスが設定される。

【0059】254において、「setrgbcolor」コマンドにより、描画図形をペイントする色を青に設定する。255において、「fill」コマンドにより、254において設定された色「青」で図6における菱形Cの内部が塗りつぶされる。青の菱形Cは、赤の第1の矩形Aに上書きされる。

【0060】256において、「restore」コマンドにより、250においてセーブされた内部状態が復元される。257において、「translate」コマンドにより、初期位置を(2048, 4160)に設定し、258において、「scale」コマンドにより、X座標の単位1を1024に、Y座標の単位1を1024にスケーリングする。259において、「triangle」により、描画図形として三角形が設定される。これは、先に定義してあるから、図4の220～226においてなされた三角形の定義におけるバスを257における初期位置と、258におけるスケーリングにしたがって変換すると、実際に描画されるべき三角形のバスが設定される。

【0061】すなわち、図6における三角形Dに示されるように、初期位置(2048, 4160)から位置(4096, 4160)まで第1のバスが設定され、位置(4096, 4160)から位置(3072, 6208)まで第2のバスが設定され、位置(3072, 6208)から初期位置(2048, 4160)までクローズバスが設定される。

【0062】260において、「setrgbcolor」コマンドにより、描画図形をペイントする色を黄に設定する。261において、「fill」コマンドにより、260において設定された色「黄」で図6における三角形Dの内部を塗りつぶす。黄の三角形Dは、緑の第2の矩形Bに上書きされる。

【0063】本発明の第1の具体例を説明する。この具体例は、入力されたコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータをそのままの形で各小領域ごとのPDLコマンドデータとするものである。

【0064】図7、図8は、本発明の第1の具体例の動作の一例の概要を示すフローチャートである。図7、図8に示されるフローに沿って、本発明の第1の具体例の動作の概要を説明する。S41～S50の処理は、PDL分割部2により実行される。S41において、内部状態がリセットされる。S42において、PDLメモリ4がリセットされる。S43において、コマンドバッファ3からPDLコマンドデータが1つずつ読み出される。

【0065】S44において、PDLコマンドデータの内容が印字であればS45に処理が進み、プリント実行であれば図8のS51に処理が進み、コマンドがなければ処理を終了する。なお、S44においては、分歧先として、印字の場合とプリント実行の場合、および終了の場合しか図示していない。しかし、PDLコマンドデータには他に種々のものがあり、これらは、例えばPDLコマンドデータをPDL分割部2のスタックメモリに記憶させておいたり、PDL分割部2の内部状態を変化させたりして、S43に処理が戻る。

【0066】S45において、PDL分割部2の内部状態や、スタックメモリに記憶されたPDLコマンドデータなどに基づいて、描画すべき画像のバスの上位置、下位置のY座標値がそれぞれ上位置、下位置レジスタにセットされる。例えば、「fill」コマンドの前にあって、先にコマンドバッファ3から読み出されてPDL分割部2のスタックメモリに記憶されたPDLコマンドデータから描画すべき画像のバスの計算が行なわれ、図3に示された画像領域上において、描画すべき画像のバスの最上点である上位置のY座標値、最下点である下位置のY座標値が計算されて、それぞれ上位置、下位置レジスタにセットされる。

【0067】S46において、初期値が設定される。小領域を指示示す小領域ポインタ1の初期値および、当該小領域の上位置のY座標値jの初期値は0である。当該小領域の下位置のY座標値kの初期値は、小領域の幅から1を引いた値であり、図3に示された画像領域の例では、小領域の幅が1024であるから、下位置のY座標値kの初期値は1023となる。

【0068】S47において、下位置のY座標値がj以上でありかつ上位置のY座標値がk以下であるか否かが判定され、下位置のY座標値がj以上でありかつ上位置のY座標値がk以下であればS48に処理が進み、そうでなければ、S49に処理が進む。描画すべき画像のバスの下位置が当該小領域の上位置jより上(Y座標値が小さい)であるか、または、描画すべき画像のバスの上位置が当該小領域の下位置kより下(Y座標値が大きい)であれば、このような画像は当該領域に描画されないものであるから、S48をスキップさせる。

【0069】したがって、入力されたPDLコマンドデータに応じた画像データが当該小領域に展開されるものであるかを検出し、当該小領域に少なくとも一部の画像データが展開されるPDLコマンドデータである場合にのみS48に処理が進む。

【0070】S48において、その内容が印字であるPDLコマンドデータおよびこれに関連するPDLコマンドデータ、例えば、「fill」コマンドの前にあって、先にコマンドバッファ3から読み出されPDL分割部2のスタックメモリに記憶されたPDLコマンドデータが、PDLメモリ4上において小領域ポインタ1に対

15

して割り当てられた領域に追加して書き込まれる。例えば、PDLメモリ4には、後述する図9、図10に示されるように、小領域の区切りをコメント行の形で書き込み、PDLコマンドデータは、小領域ポインタiに対応するコメント行の後に割り当てられ、ここに順次追加して書き込まれるようにすることができる。この場合、後述するPDLインタプリタ5は、このコメント行を判定して小領域ごとのPDLコマンドデータを理解することもできる。

【0071】なお、PDLメモリ4には、小領域ごとのPDLコマンドデータがそのまま書き込まれるようにしたが、中間コードのような形に一旦変更して書き込むことも可能である。本明細書では、中間コードのような形式で書き込まれたものも、PDLコマンドという。

【0072】S49において、当該小領域の次の小領域について同様の処理をするため、小領域ポインタiは1を加えられて更新され、同時に、当該小領域の上位置のY座標値j、当該小領域の下位置のY座標値kも、それぞれ小領域の幅を加えられて更新される。

【0073】S50において、更新された小領域ポインタiの値が、分割した小領域数より1つ小さい値を超えたか否かが判定され、超えればS43に処理が戻り、超えなければS47に処理が戻る。例えば用紙がA3サイズの場合に、図3に示されるように分割した小領域数は0から6までの7個であるから、更新されたiの値が6以下のときは、S47に処理が戻り、次の小領域について同様の処理が行なわれる。このようにして、同じ印字コマンドに対し、全小領域について同様の処理が終了すると、更新されたiの値が7となり、S42に処理が戻り次のPDLコマンドが読み出される。

【0074】図8のS51に処理が進むのは、S44において、PDLコマンドの内容が印刷実行の場合であった。このとき、PDLメモリ4には、分動作業後のPDLコマンドデータが記憶されている。図8に示されるS51～S59の処理は、PDLインタプリタ5により実行される。

【0075】S51において、初期値が設定される。上述した図7のS46と同様に、小領域ポインタiの初期値および、当該小領域の上位置のY座標値jの初期値は0であり、当該小領域の下位置のY座標値kの初期値は小領域の幅から1を引いた値である。mは、小領域バッファ6a、6bの一方への書き込みまたは読み出しを指示するフラグであり、例えば、m=0のとき小領域バッファ6aを指示し、m=1のとき小領域バッファ6bを指示する。mの初期値は0とする。

【0076】S52において、PDLメモリ4から小領域ごとのPDLコマンドデータが一つずつ読み出される。この小領域ごとのPDLコマンドデータの内容が印字である場合、S53に処理が進む。なお、S52においては、小領域ごとのPDLコマンドデータの内容が印

10

20

30

40

50

16

字でない場合の処理は、図示していない。この場合は、例えば、小領域ごとのPDLコマンドデータをPDLインタプリタ5のスタックメモリに記憶させるなどして、PDLインタプリタ5の内部状態を変化させ、S52に処理が戻る。

【0077】S53において、この時点のPDLインタプリタ5の内部状態、例えば、PDLインタプリタ5のスタックメモリに記憶された小領域ごとのPDLコマンドデータに基づいて、当該小領域の画像データが小領域バッファm上に展開される。PDLインタプリタ5のスタックメモリには、例えば、「fill」コマンドの前にあって、先にPDLメモリ4から読み出されて記憶されたコマンドデータが記憶されている。

【0078】なお、小領域バッファmにおいては、あらかじめ、以前書き込まれた画像データがクリアされている。当該小領域は、小領域ポインタiで指示され、当該小領域の上位置のY座標値はjであり、当該小領域の下位置のY座標値はkである。すなわち、jからkの部分を小領域バッファm上に画像データ展開する。

【0079】このPDLコマンドデータが画像展開されるとき、描画すべき画像のバスが当該小領域の範囲を超えることがあり得る。しかし、当該小領域の範囲内の画像データのみが展開されるようになる。当該小領域の範囲外の部分の画像データの作成は、これを行なわないか、行なったとしても小領域バッファ6a、6bへの展開を行なわない。また、当該小領域には、複数の印字コマンドが存在する場合もあるが、後に位置する印字コマンドに基づく画像データが、先に展開された画像データの上に上書きされる。

【0080】S54において、当該小領域におけるPDLコマンドデータの展開が終了したか否かを判定し、終了していない場合には、S52に処理を戻し、次のPDLコマンドデータを読み出し、同様の処理が行なわれる。終了した場合には、S55に処理が進む。展開が終了したか否かは、例えば、小領域ごとのPDLコマンドデータが、後述する図9、図10に示されるような、小領域ごとにコメント行が挿入されたものである場合には、次の小領域のコメント行および最後の印刷実行コマンドの検出により知ることができる。

【0081】S55において、小領域バッファ6aまたは6b中、mの値により指定されたものに書き込まれた画像データをプリントする要求をプリンタエンジン8に発行する。

【0082】S56において、当該小領域の次の小領域について同様の処理を行なうため、小領域ポインタiは1を加えられて更新され、同時に、当該小領域の上位置のY座標値j、当該小領域の下位置のY座標値kも、それぞれ小領域の幅を加えられて更新される。また、次に書き込むべき小領域バッファ6a、6bを切り換えるため、mの値も更新される。mの値の更新方法としては、

例えば、「m XOR 1」の演算を行ない、演算結果をmに記憶させることによりmの値を0と1との間で交互に切り換えることができる。

【0083】S57において、更新された小領域ポインタ1の値が、分割した小領域数より1つ小さい値を超えたか否かが判定され、超えればS43に処理が戻り、超えなければS58に処理が進む。

【0084】S58において、小領域バッファ6aまたは6b中、S55においてmの値により指定されたもののプリントが終了したか否かを判定し、終了した場合にはS52に処理が戻り、終了しない場合にはS59に処理が進む。S59において、プリント終了を待ち、終了時にS52に処理が戻る。プリントの終了はプリンタエンジン8から伝えられる。

【0085】図9、図10は、本発明の第1の具体例における小領域ごとのPDLコマンドデータの一例を示す説明図である。図11、図12は、本発明の第1の具体例における画像データ展開を説明する説明図である。

【0086】まず、図7のS48において追加される小領域ごとのPDLコマンドデータを図9、図10を用い、図4、図5も適宜参照しながら説明する。この一例では、小領域ごとのPDLコマンドデータは、図4、図9、図10の3部分からなるが、図面の都合上分けたにすぎず、一体としてPDLメモリ4に書き込まれるものである。なお、図4の211～234の定義に関するPDLコマンドデータはそのまま小領域ごとのPDLコマンドデータとしているので、図4のPDLコマンドデータについては、改めて図示しなかった。

【0087】図4の211～234に相当する定義に関するPDLコマンドデータ、および、図9の271における「scale」コマンド、272における「gsave」コマンド、図10の339における「showpage」コマンドは、小領域ごとに区分して記載されていない。図5の248、249における「initgraphics」コマンド、「scale」コマンドは、この具体例では不要であるので無視されている。その他のPDLコマンドは、図9、図10に示される273等のコメント行の後に小領域ごとに区分され、小領域ごとのPDLコマンドデータとなる。なお、PDLコマンドデータは、必ずしもコメント行で区分される必要はない。座標はページ記述言語における座標系で記載されている。

【0088】図4の211から図5までの236のコマンドデータは、小領域ごとのPDLコマンドデータにおいては、小領域ごとの区分の前である、図9の272までに書き込まれる。

【0089】図5の237～241の赤の第1の矩形Aに関するPDLコマンドデータは、領域5にのみ画像展開されるものであることがPDL分割部2において検出され、図5の237～241のPDLコマンドデータ

は、図10の319における領域5のコメント行の後に追加され321～325となる。なお、図10の320における「restore」コマンドも書き込まれている。

【0090】図5の242～247の緑の第2の矩形Bに関するPDLコマンドデータは、領域1、2、3に少なくとも一部の画像データが展開されるPDLコマンドデータであることがPDL分割部2において検出される。まず、図5の242～247のPDLコマンドデータは、図9の279における領域1のコメント行の後に追加され280～285となる。次に、図9の292における領域2のコメント行の後に追加され293～298となる。そして、図10の305における領域3のコメント行の後に追加され306～311となる。

【0091】図5の250～255の青の菱形Cに関するPDLコマンドデータは、領域4、5、6に少なくとも一部の画像データが展開されるPDLコマンドデータであることがPDL分割部2において検出される。まず、図5の250～255のPDLコマンドデータは、図9の312における領域4のコメント行の後に追加され313～318となる。次に、図10の319における領域5のコメント行の後に追加され326～331となる。そして、図10の332における領域6のコメント行の後に追加され333～338となる。

【0092】図5の256～261の黄の三角形Dに関するPDLコマンドデータは、領域0、1、2に少なくとも一部の画像データが展開されるPDLコマンドデータであることがPDL分割部2において検出される。まず、図5の256～261のPDLコマンドデータは、図9の273における領域0のコメント行の後に追加され274～278となる。なお、図5の256における「restore」コマンドは、最初の小領域である領域0においては不要であるので除かれている。次に、図9の279における領域1のコメント行の後に追加され286～291となる。そして、図9の292における小領域2のコメント行の後に追加され299～304となる。

【0093】最後に、図5の262における「showpage」コマンドは、小領域ごとのPDLコマンドデータにおいては、小領域ごとの区分の後の図10の339に書き込まれる。

【0094】図11、図12を用い、図9、図10も適宜参照しながら、図8のS53において小領域ごとのPDLコマンドデータが小領域ごとの画像データに展開される様子を説明する。

【0095】領域0においては、PDLインタプリタ5により、図4の211、220～226におけるPDLコマンドデータおよび、図9の271、272、273～278におけるPDLコマンドデータ、図10の339におけるPDLコマンドデータに基づいて、画像データ

タが展開される。黄の三角形Dが領域0に展開されるが、この領域0の範囲内の画像データのみが小領域バッファに書き込まれるようにする。

【0096】領域1においては、PDLインタプリタ5により、図4の211, 212~219, 220~226におけるPDLコマンドデータおよび、図9の271, 272, 279~291におけるPDLコマンドデータ、図10の339におけるPDLコマンドデータに基づいて、画像データが展開される。緑の第2の矩形Bが展開された後に黄の三角形Dが上書きされて展開されるが、この領域1の範囲内の画像データのみが書き込まれるようにする。

【0097】領域2においては、PDLインタプリタ5により、図4の211, 212~219, 220~226におけるPDLコマンドデータおよび、図9の271, 272, 292~304におけるPDLコマンドデータ、図10の339におけるPDLコマンドデータに基づいて、画像データが展開される。緑の第2の矩形Bが展開された後に黄の三角形Dが上書きされて展開されるが、この領域2の範囲内の画像データのみが書き込まれるようにする。

【0098】同様にして、領域3において緑の第2の矩形Bが展開され、領域4において青の菱形Cが展開され、領域5において赤の第1の矩形Aの後に青の菱形Cが上書きされて展開され、領域6において青の菱形Cが展開され、いずれの領域においても、領域の範囲内の画像データのみが書き込まれるようにする。

【0099】以上のようにして、本発明の第1の具体例においては、PDL分割部2が、入力されたPDLコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータを各小領域ごとのPDLコマンドデータとし、これをPDLメモリ4に記憶する。PDLインタプリタ5は、これを読み出して前記小領域ごとの画像データに展開して出力する。

【0100】本発明の第2の具体例を説明する。この具体例は、入力されたPDLコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータから各小領域の範囲内でのみ画像展開されるPDLコマンドデータを作成し、作成されたPDLコマンドデータを各小領域ごとのPDLコマンドデータとするものである。

【0101】図13、図14は、本発明の第2の具体例の動作の概要を示すフローチャートである。図13、図14は、全体的には第1の具体例の動作を示す図7、図8と同様である。したがって、図7、図8と同様なステップについては、同じ符号を用い説明を省略する。相違するステップは、S61~S64である。S41からS

50までの処理は、PDL分割部2により実行され、図14のS62からS59までの処理は、PDLインタプリタ5により実行される。

【0102】図13のS61においては、その内容が印字であるPDLコマンドデータおよびこれに関連するPDLコマンドデータ、例えば、印字コマンド「f i 1 1」の前にあって、先にコマンドバッファ3から読み出されPDL分割部2のスタックメモリに記憶されたPDLコマンドデータから、当該小領域の範囲内でのみ画像展開されるPDLコマンドデータが作成される。

【0103】そして、PDLメモリ4上において小領域ポインタ1に対して割り当てられた領域に追加して書き込まれる。例えば、PDLメモリ4には、後述する図24から図26に示されるように、小領域の区切りをコメント行の形で書き込み、PDLコマンドデータは、小領域ポインタ1に対応するコメント行の後の行に割り当てられ、ここに順次追加して書き込まれるようにすることができる。この場合、後述するPDLインタプリタ5は、このコメント行を判定して小領域ごとのPDLコマンドデータを理解することもできる。

【0104】なお、本発明の第2の具体例においても、PDLメモリには、小領域ごとのPDLコマンドデータがそのまま書き込まれるようになつたが、中間コードのような形に一旦変更して書き込むことも可能である。本明細書では、中間コードのような形式で書き込まれたものも、PDLコマンドという。S61における処理の詳細は、図15~図23を用いて後述する。

【0105】図13のS61において、既に小領域ごとのPDLコマンドデータが、各小領域の範囲内でのみ画像展開されるようになっている。したがって、図14のS63において、PDLインタプリタ5がPDLコマンドデータを当該小領域ごとに画像データに展開する際、展開された画像データがこの当該小領域の範囲内になるように処理する必要がないから、当該小領域の上位置のY座標値および当該小領域の下位置のY座標値kの値を必要としない。その結果、S62においてこれらj, kの初期値を設定する必要がなく、S64においてこれらj, kの値を更新する必要がない。

【0106】しかし、S62, S63, S64を第1の具体例と同じステップにしても動作する。このようにすることにより、PDL分割部2においては、例えば描画図形ごとに、小領域ごとのPDLコマンドデータとして、第1の具体例のものまたは第2の具体例のもののいずれかを任意に採用することができる。すなわち、分割作業後のPDLコマンドデータの形式を第1の具体例によるもの、第2の具体例によるもの、のいずれにすることもできる。

【0107】図15は、図13のS61における処理の詳細を示すフローチャートである。このフローチャートは、j, kの範囲に入るPDLコマンドデータを作成す

る動作を説明するものである。S 7 1において、PDLコマンドデータの印字要求が文字印字か図形印字であるかを判定する。文字印字要求である場合にはS 7 2に処理が進み、図形印字である場合にはS 7 3に処理が進む。S 7 2において、文字印字のための計算が行なわれ、PDLコマンドデータが作成されてこれがPDLメモリ4に書き込まれ、処理が終了する。S 7 2における処理の詳細は、図22、図23等を用いて後述する。

【0108】S 7 3において、初期設定がされる。 x_1 は、 x_3 を退避させるための変数である。この x_3 とは、バスの設定処理を行なうS 7 8において、展開される画像のバスが当該小領域の上位置（当該小領域内でY座標値が最も小さい値jをとる位置）のラインを通過する地点のX座標値である。 x_4 は、同じく x_3 を退避させるための変数である。この x_4 は、展開される画像のバスが当該小領域の下位置（当該小領域内でY座標値が最も大きい値kをとる位置）のラインを通過する地点のX座標値である。 x_0 、 x_1 にはNULLが入力される。また、 (x_1, y_1) には、展開される画像のバスの始端を表わすが入力される。

【0109】S 7 4において、 y_1 がj以上k以下であるか否かが判定される。j以上k以下である場合にはS 7 5に処理が進み、j以上k以下でない場合にはS 7 6に処理が進む。図7のステップS 4 6において説明したように、jは当該小領域の上位置のY座標値であり、kは当該小領域の下位置のY座標値である。展開される画像のバスの始端のY座標値 y_1 がこの範囲内に入っている場合は、この始端が当該小領域内に存在することを意味する。したがって、S 7 5に処理が進み、この始端のXY座標値 (x_1, y_1) を当該小領域における画像のバスの初期位置のXY座標値とする。なお、展開される画像のバスの始端のY座標値 y_1 がこの範囲内に入っていない場合は、初期位置は未定義状態とし、後に行なわれるS 7 8におけるバスの設定動作の中で初期位置が設定されることとなる。

【0110】S 7 6において、PDL分割部2の例えは、スタックメモリに書き込まれたPDLコマンドを1つずつ読み出し、S 7 7においてこのPDLコマンドの種類を判定する。PDLコマンドが「line to」等の印字バスの設定コマンドである場合はS 7 8に処理を進め、「transfer」等の座標変換コマンドである場合はS 7 9に処理を進め、その他のPDLコマンド、例えば、色設定コマンド「set color」等である場合にはS 8 0に処理を進める。また、「fill」等の図形印字コマンドである場合はS 8 1に処理を進める。

【0111】S 7 8においては、印字バスの設定コマンドに基づいて印字バスの計算が行なわれる。S 7 9においては、座標変換コマンドに基づいて内部的な座標系の変換が行なわれるが、この座標変換コマンドのPDLメ

モリ4への出力は行なわれない。S 8 0においては、PDLコマンドがそのまま実行されるようにこのPDLコマンドがPDLメモリ4に設定される。S 7 8～S 8 0では、処理がS 7 9に戻り、次のPDLコマンドが読み出されるようにされる。S 8 1においては、図形印字コマンドが実行されるようにこのPDLコマンドがPDLメモリ4に設定されて処理が終了する。

【0112】図16は、図15のS 7 8における処理の詳細を示すフローチャートである。S 9 1においては、印字バスの設定コマンドに基づいて印字バスの計算が行なわれ、このバスの各部の位置が計算される。バスの上位値の座標値は (x_1, y_1) に入力され、バスの下位値は (x_2, y_2) に入力される。バスの方向はdに入力される。

【0113】ここで、バスの上位値とは、図3に示された画像領域上において、1本のバスの最上点が上位置であり、最下点が下位置である。また、バスの方向dとは、1本のバスの始端が上位置でその終端が下位置であるか、逆に、1本のバスの始端が下位置でその終端が上位置であるかを示すフラグである。すなわち、前者の場合は、バスの方向が上から下であり、後者の場合は、バスの方向が下から上である。

【0114】なお、バスが、円弧、橢円弧等のように、1本のバスの上位置、下位置と、1本のバスの始端、終端とが必ずしも対応しない場合がある。すなわち、1本のバスの始端と終端の中間に上位置および下位置の少なくとも一方が存在する場合がある。このような場合に対処するため、このようなバスは、あらかじめ、上位置または下位置においてバスを分解し、分解されたバスについてバスの設定処理を行なえばよい。

【0115】S 9 2において、バスの上位値のY座標値 y_1 、バスの下位値のY座標値 y_2 、当該小領域の上位置のY座標値j、当該小領域の下位置のY座標値kの相互の上下関係を判定することにより、印字バスの設定コマンドに基づいて計算された描画すべき画像のバスの位置と当該小領域の範囲との位置関係を判定する。そして、この位置関係に応じ、それぞれ異なるバスの計算処理をさせることにより、コマンドバッファ3に書き込まれたPDLコマンドが各小領域の範囲内において画像データに展開されるようなPDLコマンドに変換されて小領域ごとに分割されたPDLコマンドデータが作成される。

【0116】 $(y_1 < j) \text{ AND } (y_2 < j)$ か、または、 $(y_1 > k) \text{ AND } (y_2 > k)$ であった場合、バスは、当該小領域の範囲外にある。したがって、このバスは当該小領域の範囲外においてバスの先端が移動するだけで、当該小領域の範囲内への印字は全く行なわれず、結局、このバスは無視され、処理が終了する。

【0117】なお、図13のS 4 7において、このよう

なバスについては、S 6 1すなわちこの図16の処理が

されないように分岐するから、このようなバスは検出されない。したがって、図13のS45～S47の処理は、省略することも可能である。

【0118】 $(y_1 < j) \text{ AND } (y_2 \geq j) \text{ AND } (y_2 < k)$ であった場合、バスは、当該小領域の上位置(Y座標値j)のラインのみをまたぐ。S93に処理が進み、バスの計算1が行なわれて処理が終了する。

【0119】 $(y_1 \geq j) \text{ AND } (y_2 \leq k)$ であった場合、バスは、当該小領域の上位置(Y座標値j)のライン、当該小領域の下位置(Y座標値k)のラインの両方をまたがず、当該小領域の範囲内にある。この場合は、S94に処理が進みバスの計算2が行なわれて処理が終了する。

【0120】 $(y_1 \geq j) \text{ AND } (y_1 \leq k) \text{ AND } (y_2 > k)$ であった場合、バスは、当該小領域の下位置(Y座標値k)のラインのみをまたぐ。この場合は、S95に処理が進み、バスの計算3が行なわれて処理が終了する。

【0121】 $(y_1 < j) \text{ AND } (y_2 > k)$ であった場合、バスは、当該小領域の上位置(Y座標値j)のライン、当該小領域の下位置(Y座標値k)のラインの両方をまたぐ。この場合は、S96に処理が進みバスの計算4が行なわれて処理が終了する。

【0122】図17は、図16のS93における処理の詳細を示すフローチャートである。バスは、当該小領域の上位置(Y座標値j)のラインのみをまたぐが、この場合、3つのケースがある。第1のケースは、初めて上位置のラインをまたぐ場合、第2のケースは、上位置側から入ってきたバスが、再び上位置側に戻る場合、第3のケースは、逆に上位置側に出ていたバスが再び戻って来る場合である。

【0123】第1のケースでは、バスが上位置(Y座標値j)のラインをまたいだ地点を記憶しておき、小領域の範囲内のバスを設定する。このとき、まだ初期位置が設定されていないときには、バスが上位置(Y座標値j)のラインをまたいだ地点を初期位置として設定する。第2のケースでは、小領域範囲内のバスを設定した後、記憶しておいた、バスが以前に通過した地点までのバスを作成し、設定を行なう。このとき、バスが以前に通過した地点が初期位置であるなら「closepath」設定を行なう。第3のケースでは、バスが以前に通過した地点から、今回のまたいだ地点までのバスを作成し、そのバスの設定を行なった後に、小領域範囲内のバスの設定を行なう。

【0124】S101において、バスが当該小領域の上位置(Y座標値j)のラインを通過する地点のX座標値が x_1 に入力され記憶される。S102において、当該小領域において初期位置が設定済みか否かが判定される。設定済みであった場合には、S104に処理が進み、設定済みでなかった場合には、S103において、

バスが当該小領域の上位置(Y座標値j)のラインを通過する地点 (x_1, j) を初期位置として設定し、処理がS104に進む。

【0125】S104において、 x_1 がNu11か否かが判定される。Nu11であった場合にはS105に処理が進み、Nu11でなかった場合にはS106に処理が進む。 x_1 は、バスが当該小領域の上位置(Y座標値j)のラインを通過する地点のX座標値 x_1 を退避させるための変数である。最初は図15のS73においてNu11が入力され、S105に処理が進むが、後述するS109において、この時点の x_1 を退避させ、第2、第3のケースの場合に、再度バスが当該小領域の上位置(Y座標値j)のラインを通過するときに備える。そして、再度バスが通過するとき、S106に処理が進む。なお、再度のバスが通過後、後述するS115において、 x_1 にNu11が入力されるから、3度目以降の通過に対して、S104は、1回目、2回目の場合と同様の判定をすることになる。

【0126】S105に処理が進むのは、バスが初めて上位置のラインをまたぐ第1のケースである。S105において、バスの方向dが上から下か否かが判定される。上から下であった場合にはS107に、下から上であった場合にはS108に処理が進む。S107において、 (x_1, j) から (x_2, y_2) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。S108においても同様に、 (x_1, y_2) から (x_2, j) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。S107、S108のいずれも処理を終了するとS109に処理が進み、 x_1 に x_2 が入力され x_1 の値が退避され、処理が終了する。

【0127】S106において、バスの方向dが上から下か否かが判定される。上から下であった場合にはS110に、下から上であった場合にはS111に処理が進む。

【0128】S111に処理が進むのは、上位置側から入ってきたバスが再び上位置側に戻る第2のケースである。S111において、 (x_1, y_2) から (x_1, j) へのバスの設定が行なわれる。このバスは、当該小領域の範囲内のバスそのものである。次に、S112において、 (x_1, j) が初期位置であるか否かが判定される。初期位置であった場合にはS113に処理が進み、初期位置でなかった場合にはS114に処理が進む。

【0129】S113においては、「closepath」の設定が行なわれ、このPDLコマンドが作成される。S114においては、 (x_1, j) から (x_1, j) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。いずれのバスも、バスが上位置側から入ったときのライン通過地点から今回ライン

25

をまたぐ地点までのものである。 (x_i, j) が初期位置であれば、この初期位置に向けて「closepath」コマンドでバスをつなぎ、 (x_i, j) が初期位置でなければ、このバスに対応するPDLコマンドでバスをつなぐ。S113, S114のいずれもS115に処理が進み、S115において、 x_i にNullが入力され、次回にバスが当該小領域の上位置 (Y座標値 j) のラインを通過する場合は、第1のケースとして扱われるようにして、処理が終了する。

【0130】S110に処理が進むのは、上位置側に出でていたバスが再び戻って来る第3のケースである。S110において、 (x_i, j) から (x_i, j) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、バスが上位置側に出たときのライン通過地点から今回ラインをまたぐ地点までをつなぐバスである。次にS116において、 (x_i, j) から (x_i, y_2) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、当該小領域の範囲内のバスそのものである。次に、S115において、 x_i にNullが入力され処理が終了する。

【0131】図18は、図16のS94における処理の詳細を示すフローチャートである。バスは、当該小領域の範囲内にあり、この範囲の中だけで閉じている。この場合は単に元のPDLコマンドが作成される。S121において、バスの方向dが上から下か否かが判定される。上から下であった場合にはS122に、下から上であった場合にはS123に処理が進む。S122において、 (x_i, y_1) から (x_i, y_2) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成され処理が終了する。S123においても同様に、 (x_i, y_2) から (x_i, y_1) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成され処理が終了する。

【0132】図19は、図16のS95における処理の詳細を示すフローチャートである。バスは、当該小領域の下位置 (Y座標値 k) のラインのみをまたぐが、この場合も、バスの計算1の場合と同様に、3つのケースがある。第1のケースは、初めて下位置のラインをまたぐ場合、第2のケースは、下位置側から入ってきたバスが、再び下位置側に戻る場合、第3のケースは、逆に下位置側に出ていたバスが再び戻って来る場合である。

【0133】第1のケースでは、バスが下位置 (Y座標値 k) のラインをまたいだ地点を記憶しておき、小領域の範囲内のバスを設定する。このとき、まだ初期位置が設定されていないときには、バスが下位置 (Y座標値 k) のラインをまたいだ地点を初期位置として設定する。第2のケースでは、小領域範囲内のバスを設定した後、記憶しておいた、バスが以前に通過した地点までのバスを作成し、設定を行なう。このとき、バスが以前に

10

20

30

40

50

26

通過した地点が初期位置であるなら「closepath」設定を行なう。第3のケースでは、バスが以前に通過した地点から、今回のまたいだ地点までのバスを作成し、そのバスの設定を行なった後に、小領域範囲内のバスの設定を行なう。

【0134】S131において、バスが当該小領域の下位置 (Y座標値 k) のラインを通過する地点のX座標値が x_i に入力され記憶される。S132において、当該小領域において初期位置が設定済みか否かが判定される。設定済みであった場合には、S134に処理が進み、設定済みでなかった場合には、S133において、バスが当該小領域の下位置 (Y座標値 k) のラインを通過する地点 (x_i, k) を初期位置として設定し、処理がS134に進む。

【0135】S134において、 x_i がNullか否かが判定される。Nullであった場合にはS135に処理が進み、Nullでなかった場合にはS136に処理が進む。 x_i は、バスが当該小領域の下位置 (Y座標値 k) のラインを通過する地点のX座標値 x_i を退避させるための変数である。最初は図15のS73においてNullが入力され、S135に処理が進むが、後述するS139において、この時点の x_i を退避させ、第2、第3のケースの場合に、再度バスが当該小領域の下位置 (Y座標値 k) のラインを通過するときに備える。そして、再度バスが通過するとき、S136に処理が進む。なお、再度のバスが通過後、後述するS143において、 x_i にNullが入力されるから、3度目以降の通過に対しては、S134は、1回目、2回目の場合と同様の判定をすることになる。

【0136】S135に処理が進むのは、バスが初めて下位置のラインをまたぐ第1のケースである。S135において、バスの方向dが上から下か否かが判定される。上から下であった場合にはS137に、下から上であった場合にはS138に処理が進む。S137において、 (x_i, y_1) から (x_i, k) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。S138においても同様に、 (x_i, k) から (x_i, y_1) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。S137, S138のいずれも処理を終了するとS139に処理が進み、 x_i に x_i が入力され x_i の値が退避され、処理が終了する。

【0137】S136において、バスの方向dが上から下か否かが判定される。上から下であった場合にはS140に、下から上であった場合にはS141に処理が進む。

【0138】S140に処理が進むのは、下位置側から入ってきたバスが再び下位置側に戻る第2のケースである。S140において、 (x_i, y_1) から (x_i, k) へのバスの設定が行なわれる。このバスは、当該小

領域の範囲内のバスそのものである。次に、S 1 4 4において、 (x_1, k) が初期位置であるか否かが判定される。初期位置であった場合には S 1 4 5 に処理が進み、初期位置でなかった場合には S 1 4 6 に処理が進む。

【0 1 3 9】 S 1 4 5 においては、「closepath」の設定が行なわれ、このPDLコマンドが作成される。S 1 4 6 においては、 (x_1, k) から (x_1, k) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。いずれのバスも、バスが下位置側から入ったときのライン通過地点から今回ラインをまたぐ地点までのものである。 (x_1, k) が初期位置であれば、この初期位置に向けて「closepath」コマンドでバスをつなぎ、 (x_1, k) が初期位置でなければ、このバスに対応するPDLコマンドでバスをつなぐ。S 1 4 5, S 1 4 6 のいずれも S 1 4 3 に処理が進み、S 1 4 3 において、 x_1 に Null が入力され、次回にバスが当該小領域の下位置 (Y座標値 k) のラインを通過する場合は、第 1 のケースとして扱われるようにして、処理が終了する。

【0 1 4 0】 S 1 4 1 に処理が進むのは、下位置側に出ていたバスが再び戻って来る第 3 のケースである。S 1 4 1 において、 (x_1, k) から (x_1, k) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、バスが下位置側に出たときのライン通過地点から今回ラインをまたぐ地点までをつなぐバスである。次に S 1 4 2 において、 (x_1, k) から (x_1, y_1) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、当該小領域の範囲内のバスそのものである。次に、S 1 4 3 において、 x_1 に Null が入力され、処理が終了する。

【0 1 4 1】 図 2 0, 図 2 1 は、図 1 6 の S 9 6 における処理の詳細を示すフローチャートである。バスは、当該小領域の上位置 (Y座標値 j) のライン、当該小領域の下位置 (Y座標値 k) のラインの両方をまたぐ。この場合は、バスが当該小領域の上位置 (Y座標値 j) のラインをまたぐバスの計算 1 の動作と、バスが当該小領域の下位置 (Y座標値 k) のラインをまたぐバスの計算 3 の動作とを組み合わせたものとなる。すなわち、それぞれのラインをまたぐ地点を計算して、それをバスとする。

【0 1 4 2】 バスが上から下であり、初期位置がまだ設定されていなければ、バスが上位置 (Y座標値 j) のラインをまたいだ地点を初期位置とする。また、前にバスが上位置 (Y座標値 j) のラインをまたいだのであれば、その地点から本バスが上位置 (Y座標値 j) のラインをまたぐ地点までのバスを本バスの前に加える。また、前にバスが下位置 (Y座標値 k) のラインをまたいだのであれば、本バスが下位置 (Y座標値 k) のライン

をまたぐ地点から前にバスが下位置 (Y座標値 k) のラインをまたいだ地点までのバスを本バスの後に加える。バスが下から上にいく場合も同様に、バスの計算を行なう。

【0 1 4 3】 この場合、6つのケースがある。第 1 のケースは、バスが初めて上位置と下位置のラインをこの順でまたぐ場合、第 2 のケースは、上位置側に出ていたバスが再び戻って来て上位置のラインと下位置のラインをこの順でまたぐ場合、第 3 のケースは、下位置から入ってきたバスが上位置側に出て行き、再び戻って来て上位置のラインと下位置のラインとをこの順にまたぐ場合、第 4 のケースは、バスが初めて下位置と上位置のラインをこの順でまたぐ場合、第 5 のケースは、下位置側に出ていたバスが再び戻って来て下位置のラインと上位置のラインをこの順でまたぐ場合、第 6 のケースは、上位置から入ってきたバスが下位置側に出て行き、再び戻って来て下位置のラインと上位置のラインとをこの順にまたぐ場合である。

【0 1 4 4】 S 1 5 1 において、バスが当該小領域の上位置 (Y座標値 j) のラインを通過する地点の X 座標値が x_1 に入力され、バスが当該小領域の下位置 (Y座標値 k) のラインを通過する地点の X 座標値が x_1 に入力され記憶される。S 1 5 2 において、バスの方向 d が上から下か否かが判定される。上から下であった場合には S 1 5 3 に、下から上であった場合には図 2 1 の S 1 7 3 に処理が進む。

【0 1 4 5】 S 1 5 3 に処理が進むのは、第 1 ～ 第 3 のケースである。S 1 5 3 において、当該小領域において初期位置が設定済みか否かが判定される。設定済みであった場合には、S 1 5 5 に処理が進み、設定済みでなかった場合には、S 1 5 4 において、バスが当該小領域の上位置 (Y座標値 j) のラインを通過する地点 (x_1, j) を初期位置として設定し、処理が S 1 5 5 に進む。

【0 1 4 6】 S 1 5 5 において、 x_1 が Null か否かが判定される。Null であった場合には S 1 5 6 に処理が進み、Null でなかった場合には S 1 5 7 に処理が進む。 x_1 には、最初、図 1 5 の S 7 3 において Null が入力され、S 1 5 6 に処理が進むが、後述する S 1 6 0 において、この時点の x_1 を退避させ、第 2, 第 3 のケースの場合に、再度バスが当該小領域の上位置 (Y座標値 j) のラインを通過するときに備えるものである。そして、再度バスが通過するとき、S 1 5 7 に処理が進む。なお、再度のバスが通過後、後述する S 1 5 9 において、 x_1 に Null が入力されるから、3度目以降の通過に対しては、S 1 5 5 は、1回目、2回目の場合と同様の判定をすることになる。

【0 1 4 7】 S 1 5 6 に処理が進むのは、第 1 のケースである。S 1 5 6 において、 (x_1, j) から (x_1, k) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成され、S 1 6 0 において、 x_1 に x

が入力され x_1 の値が退避され、S 161 に処理が進む。

【0148】S 157 に処理が進むのは、第2、第3のケースである。S 157において、 (x_1, j) から (x_1, j) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、バスが上位置側に出たときのライン通過地点から今回ラインをまたぐ地点までをつなぐバスである。次にS 158において、 (x_1, j) から (x_1, k) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、当該小領域の範囲内のバスそのものである。以上の2つのバスの設定が行なわれる。次に、S 159において、 x_1 にNu 11が入力され、次回にバスが当該小領域の上位置(Y座標値 j)のラインを通過する場合は、第4のケースとして扱われるようにして、S 161 に処理が進む。

【0149】S 161において、 x_1 がNu 11か否かが判定される。Nu 11であった場合にはS 162に処理が進み、Nu 11でなかった場合にはS 163に処理が進む。 x_1 には、最初、図15のS 73においてNu 11が入力され、S 162に処理が進む。

【0150】S 162に処理が進むのは、第1、第2のケースである。S 162において、この時点の x_1 を退避させる。

【0151】S 163に処理が進むのは、第3のケースである。S 163においては、 (x_1, k) が初期位置であるか否かが判定される。初期位置であった場合にはS 164に処理が進み、初期位置でなかった場合にはS 166に処理が進む。S 164においては、「c los e path」の設定が行なわれ、このPDLコマンドが作成される。S 166においては、 (x_1, k) から (x_1, k) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。いずれのバスも、バスが下位置側から入ったときのライン通過地点から今回ラインをまたぐ地点までのものである。 (x_1, k) が初期位置であれば、この初期位置に向けて「c los e path」コマンドでバスをつなぎ、 (x_1, j) が初期位置でなければ、このバスに対応するPDLコマンドでバスをつなぐ。

【0152】S 164、S 166の終了後は、いずれもS 165に処理が進み、S 165において、 x_1 にNu 11が入力され、次回にバスが当該小領域の下位置(Y座標値 k)のラインをまたぐ場合は、第4のケースとして扱われるようにして、処理が終了する。

【0153】S 173に処理が進むのは、第4～第6のケースである。S 173において、当該小領域において初期位置が設定済みか否かが判定される。設定済みであった場合には、S 175に処理が進み、設定済みでなかった場合には、S 174において、バスが当該小領域の下位置(Y座標値 k)のラインを通過する地点 (x_1, j) を初期位置として設定し、処理がS 175に進む。

【0154】S 175において、 x_1 がNu 11か否かが判定される。Nu 11であった場合にはS 176に処理が進み、Nu 11でなかった場合にはS 177に処理が進む。 x_1 には、最初、図15のS 73においてNu 11が入力され、S 176に処理が進むが、後述するS 180において、この時点の x_1 を x_1 に退避させ、第5、第6のケースの場合に、再度バスが当該小領域の下位置(Y座標値 k)のラインを通過するときに備えるものである。そして、再度バスが通過するとき、S 177に処理が進む。なお、再度のバスが通過後、後述するS 179において、 x_1 にNu 11が入力されるから、3度目以降の通過に対しては、S 175は、1回目、2回目の場合と同様の判定をすることになる。

【0155】S 176に処理が進むのは、第4のケースである。S 176において、 (x_1, k) から (x_1, j) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成され、S 180において、 x_1 に x_1 が入力され x_1 の値が退避され、S 181に処理が進む。

【0156】S 177に処理が進むのは、第5、第6のケースである。S 177において、 (x_1, k) から (x_1, k) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、バスが下位置側に出たときのライン通過地点から今回ラインをまたぐ地点までをつなぐバスである。次にS 178において、 (x_1, k) から (x_1, j) へのバスの設定が行なわれ、このバスに対応するPDLコマンドが作成される。このバスは、当該小領域の範囲内のバスそのものである。以上の2つのバスの設定が行なわれる。次に、S 179において、 x_1 にNu 11が入力され、次回にバスが当該小領域の下位置(Y座標値 k)のラインを通過する場合は、第4のケースとして扱われるようにして、S 181に処理が進む。

【0157】S 181において、 x_1 がNu 11か否かが判定される。Nu 11であった場合にはS 182に処理が進み、Nu 11でなかった場合にはS 183に処理が進む。 x_1 には、最初、図15のS 73においてNu 11が入力され、S 182に処理が進む。

【0158】S 182に処理が進むのは、第4、第5のケースである。S 182において、この時点の x_1 を退避させる。

【0159】S 183に処理が進むのは、第6のケースである。S 183においては、 (x_1, j) が初期位置であるか否かが判定される。初期位置であった場合にはS 184に処理が進み、初期位置でなかった場合にはS 186に処理が進む。S 184においては、「c los e path」の設定が行なわれ、このPDLコマンドが作成される。S 186においては、 (x_1, j) から (x_1, j) へのバスの設定が行なわれ、このバスに対

応するPDLコマンドが作成される。いずれのバスも、バスが下位置側に出たときのライン通過地点から今回ラインをまたぐ地点までのものである。 (x_0, j) が初期位置であれば、この初期位置に向けて「closepath」コマンドでバスをつなぎ、 (x_1, j) が初期位置でなければ、このバスに対応するPDLコマンドでバスをつなぐ。

【0160】S184, S186の終了後は、いずれもS185に処理が進み、S185において、 x_0 にNu11が入力され、次回にバスが当該小領域の下位置(Y座標値k)のラインをまたぐ場合は、第1のケースとして扱われるようにして、処理が終了する。

【0161】図22、図23は、図15のS72における処理の詳細を示すフローチャートである。このフローチャートは、印字要求が文字であった場合のPDLコマンドデータの作成を行なうものである。

【0162】S190において、初期設定がされる。current行には1が入力され、第1行目から始められる。PutStart, PutCompleteにはFALSEが入力される。また、当該小領域の範囲でクリップするようにクリップバスの設定が行われる。S191において、PDL分割部2の例えはスタックメモリに書き込まれたPDLコマンドを1つずつ読み出し、S192においてこのPDLコマンドの種類を判定する。

【0163】PDLコマンドが「moveto」など初期位置の設定命令であった場合、このコマンドを無視し、S191に処理が戻る。初期位置は、後で文字の印字コマンド作成時に設定される。PDLコマンドが「show」など文字印字コマンドであった場合、S193に処理が進み、印字する文字の設定を行ない、設定終了時点で終了する。PDLコマンドが他のコマンド、例えは、色設定命令「setcolor」などであった場合、S194に処理が進み、S194においては、PDLコマンドがそのまま実行されるようにこのPDLコマンドがPDLメモリ4に設定される。

【0164】S193において、 (x_1, y_1) に現在行の上位置(Y座標値が小さい側の位置)の座標値を入力し、 (x_2, y_2) に現在行の下位置(Y座標値が大きい側の位置)の座標値を入力する。S195において、まず、現在行の下位置のY座標値 y_2 が当該小領域の上位置(Y座標値j)より上側にあるか否かを判定する。上側にあった場合はS196に処理を進め、PutCompleteにTRUEを入力し、図23のS197に処理が進む。下側にあった場合は、この行の処理は飛ばされ、直接S197に処理が進む。

【0165】S197において、現在行の上位置のY座標値 y_1 が当該小領域の下位置(Y座標値k)より上側にあるか否かを判定する。上側にあった場合はS198に処理が進み、下側にあった場合は、まだ印字を行なう必要がないからS202に処理が進む。

【0166】S198において、PutStartがFALSEか否かが判定され、FALSEであった場合には、初期位置がまだ設定されていないから、S199に処理が進み、FALSEでなかった場合には、S200に処理が進む。S199において、 (x_1, y_1) を初期位置と設定し、PutStartにTRUEを入力し、S200に処理が進む。

【0167】S200において、PutCompleteがFALSEか否かが判定される。FALSEであった場合は、S201に処理が進み、FALSEでなかった場合は、S202に処理が進む。S201に処理が進むのは、現在行の領域が当該小領域の一部にでもかかる場合であり、現在行の印字が行なわれるようになる。S201において、現在行の印字を設定し、対応するPDLコマンドが作成され、S202に処理が進む。

【0168】S202において、current行は1を加えられ、現在行が更新される。S203において、文字印字コマンドの処理が終了したか否かが判定され、終了した場合は、S204に処理が進み、終了しない場合には、S193に処理が戻り、次の行について処理がなされる。このようにして、当該小領域の範囲において、全ての行に対して同様の処理が繰り返される。最後にS204において、文字印字コマンドが設定されPDLメモリ4に書き込まれる。このとき、初期位置設定コマンド等も同時に設定される。

【0169】図24、図25、図26は、本発明の第2の具体例における小領域ごとのPDLコマンドデータの一例を示す説明図である。また、図27、図28は、本発明の第2の具体例における画像データ展開を説明する説明図である。

【0170】まず、図13のS61において作成される小領域ごとのPDLコマンドデータを、図4、図5を適宜参照しながら図24、図25、図26を用いて説明する。この一例では、小領域ごとのPDLコマンドデータは、図4、図24、図25、図26の4部分からなるが、単に図面の都合上分けたにすぎず、一体としてPDLメモリ4に書き込まれるものである。なお、図4の211～234の定義に関するPDLコマンドデータは、そのまま小領域ごとのPDLコマンドデータとしているので、図4のPDLコマンドデータについては、改めて図示しなかった。

【0171】第1の具体例における小領域ごとのPDLコマンドデータと同様に、小領域ごとに区分して記載されていないPDLコマンドデータがある。図5の248、249における「initgraphics」コマンド、「scale」コマンドは、この具体例では不要であるので無視されている。その他のPDLコマンドは、図9、図10に示される273等のコメント行の後ろに小領域ごとに区分され、小領域ごとのPDLコマンドデータとなる。なお、PDLコマンドデータは、必ず

しもコメント行で区分される必要はない。座標はページ記述言語における座標系で記載されている。

【0172】図4の211から図5の236までのコマンドデータは、小領域ごとのPDLコマンドデータにおいては、小領域ごとの区分の前である、図24の342までに記載される。しかし、第2の具体例においては、図4に示される212～234における定義コマンドを用いて、図5の235～262におけるPDLコマンドデータから、新たにPDLコマンドが作成され、これが小領域ごとのPDLコマンドデータとされるから、図4に示される212～234中の使用済みの定義コマンドは必ずしもPDLメモリ4に書き込む必要がない。

【0173】図5の237～241の赤の第1の矩形Aに関するPDLコマンドデータは、領域5にのみ画像展開されるものであることがPDL分割部2において検出される。237における「translate」コマンド、238における「scale」コマンドにより、PDL分割部2の内部状態が変化する。239における「square」は、先に定義されているから、図4の212～219においてなされた矩形の定義におけるバスを237における初期位置設定と、238におけるスケーリングにしたがって変換すると、実際に描画されるべき赤の第1の矩形Aのバスが設定される。

【0174】この赤の第1の矩形Aは領域5の範囲内にのみ存在するから、この実際に描画されるべき赤の第1の矩形Aのバスのそれぞれに対応するPDLコマンドデータが作成され、図26の408における領域5のコメント行の後に書き込まれ410～415となる。なお、「restore」コマンドも書き込まれて409となる。図5の240における「setrgbcolor」コマンド、241における「fill」コマンドも書き込まれて416、417となる。

【0175】図5の242～247の緑の第2の矩形Bに関するPDLコマンドデータは、領域1、2、3に画像展開されるものであることがPDL分割部2において検出される。243における「translate」コマンド、244における「scale」コマンドにより、PDL分割部2の内部状態が変化する。239における「square」は、先に定義されているから、図4の212～219においてなされた矩形の定義におけるバスを243における初期位置設定と、244におけるスケーリングにしたがって変換すると、実際に描画されるべき緑の第2の矩形Bのバスが設定される。

【0176】第1のバスは初期位置(1024, 3136)から位置(1024, 5184)まで、第2のバスは位置(1024, 5184)から位置(3072, 5184)まで、第3のバスは位置(3072, 5184)から位置(3072, 3136)まで、第4のバスは位置(3072, 3136)から初期位置(1024, 3136)までである。

【0177】まず、領域1において作成されるPDLコマンドデータについて説明する。第1のバスは、領域1の下位置を下から上にまたぐものであり、またぐ地点は、位置(1024, 4673)であるから、初期位置を(1024, 4673)とするための「moveto」コマンドと、位置(1024, 5184)までの「lineto」コマンドが作成され、図24の351における領域1のコメント行の後に書き込まれて354, 355となる。第2のバスは、領域1の範囲内のものであるから、「lineto」コマンドが作成され、図24の355の次に書き込まれて356となる。第3のバスは、領域1の下位置を上から下にまたぐものである。またぐ地点は、位置(3072, 4673)であるから、この位置までの「lineto」コマンドが作成されるとともに、この位置から初期位置までの「closepath」コマンドが作成され、図24の356の後に書き込まれて357, 358となる。第4のバスは、領域1の範囲外であるためPDLコマンドは作成されない。なお、図5の242における「restore」コマンド、図4の213における「newpath」コマンドも書き込まれて352, 353となる。図5の246における「setrgbcolor」コマンド、247における「fill」コマンドも書き込まれて359, 360となる。

【0178】領域2において作成されるPDLコマンドデータについて説明する。第1のバスは、領域2の下位置を下から上にまたぎ、かつ上位置も下から上にまたぐものである。下位置をまたぐ地点は、位置(1024, 3649)であり、上位置をまたぐ地点は、位置(1024, 4672)であるから、初期位置を(1024, 3649)とするための「moveto」コマンドと、位置(1024, 4672)までの「lineto」コマンドが作成され、図25の370における領域2のコメント行の後に書き込まれて373, 374となる。第2のバスは、領域2の範囲外のものであるから、PDLコマンドは作成されない。第3のバスは、領域2の上位置を上から下にまたぎ、下位置も上から下にまたぐものである。上位置をまたぐ地点は、位置(3072, 4672)であり、下位置をまたぐ地点は、位置(3072, 3649)であるから、位置(3072, 4672)までの「lineto」コマンドと、位置(3072, 3649)までの「lineto」コマンドと、初期位置までの「closepath」コマンドが作成され、図25の374の後に書き込まれて375, 376, 377となる。第4のバスは、領域2の範囲外のものであるからPDLコマンドは作成されない。なお、領域1の場合と同様に、「restore」コマンド、「newpath」コマンド、「setrgbcolor」コマンド、「fill」コマンドも書き込まれて371, 372, 378, 379となる。

【0179】領域3において作成されるPDLコマンドデータについて説明する。第1のバスは、その始端が領域3の範囲内の位置(1024, 3136)にあるものであり、かつ、領域3の上位置を下から上にまたぐものである。また地点は、位置(1024, 3648)である。したがって、初期位置を(1024, 3136)とする「moveto」コマンドが作成されるとともに、(1024, 3648)までの「lineto」コマンドが作成され、図25の389における領域3のコメント行の後に書き込まれて392, 393となる。第2のバスは、領域3の範囲外のものであり、PDLコマンドは作成されない。第3のバスは、領域3の上位置を上から下にまたぐものである。また地点は、位置(3072, 3648)であるから、この位置までの「lineto」コマンドが作成されるとともに、位置(3072, 3136)までの「lineto」コマンドが作成され、図25の393の後に書き込まれて394, 395となる。第4のバスは、領域3の範囲内のものであり、初期位置までの「closepath」コマンドが作成され、図25の395の後に書き込まれて396となる。なお、領域1の場合と同様に、「grestore」コマンド、「newpath」コマンド、「setrgbcolor」コマンド、「fill」コマンドも書き込まれて390, 391, 397, 398となる。

【0180】図5の250から255までの青の菱形Cに関するPDLコマンドデータは、領域4, 5, 6に画像展開されるものであることがPDL分割部2において検出される。251における「translate」コマンド、252における「scale」コマンドにより、PDL分割部2の内部状態が変化する。253における「diamond」コマンドは、先に定義されているコマンドであるから、図4の227～234においてなされた菱形の定義におけるバスを251における初期位置設定と、252におけるスケーリングにしたがって変換すると、実際に描画されるべき青の菱形Cのバスが設定される。

【0181】第1のバスは初期位置(1024, 1088)から位置(2560, 64)まで、第2のバスは位置(2560, 64)から位置(4096, 1088)まで、第3のバスは位置(4096, 1088)から位置(2560, 2112)まで、第4のバスは位置(2560, 2112)から初期位置(1024, 1088)までである。

【0182】まず、領域4において作成されるPDLコマンドデータについて説明する。第1, 第2のバスは、領域4の範囲外のものであるから、PDLコマンドデータを作成しない。第3のバスは、領域4の下位置を下から上にまたぐものであり、また地点は、位置(3327, 1601)であるから、初期位置を(3327, 1601)とするための「moveto」コマンドと、位

置(2560, 2112)までの「lineto」コマンドが作成され、図25の399における領域4のコメント行の後に書き込まれて402, 403となる。第4のバスは、領域1の下位置を上から下にまたぐ。また地点は、位置(1793, 1601)であるから、この位置までの「lineto」コマンドが作成されるとともに、この位置から初期位置までの「closepath」コマンドが作成され、図25の403の後に書き込まれて404, 405となる。他の図形の場合と同様に、「grestore」コマンド、「newpath」コマンド、「setrgbcolor」コマンド、「fill」コマンドも書き込まれて400, 401, 406, 407となる。

【0183】領域5において作成されるPDLコマンドデータについて説明する。第1のバスは、その始端が領域5の範囲内の位置(1024, 1088)にあるものであり、かつ、領域5の下位置を上から下にまたぐものである。下位置をまたぐ地点は、位置(1791, 577)である。したがって、初期位置を(1024, 1088)とするための「moveto」コマンドと、位置(1791, 577)までの「lineto」コマンドが作成され、図26の408における領域5のコメント行の後に書き込まれて420, 421となる。第2のバスは、領域5の下位置を下から上にまたぐものである。また地点は、位置(3329, 577)であるから、この位置までの「lineto」コマンドが作成されるとともに、(4096, 1088)までの「lineto」コマンドが作成され、図26の421の後に書き込まれて、422, 423となる。第3のバスは、領域5の上位置を下から上にまたぐものである。また地点は、位置(3328, 1600)であるから、この位置までの「lineto」コマンドが作成され、図26の423の後に書き込まれて、424となる。第4のバスは、領域5の上位置を上から下にまたぐものである。また地点は、位置(1792, 1600)であるから、この位置までの「lineto」コマンドが作成されるとともに、初期位置までの「closepath」コマンドが作成され、図26の424の次の425, 426に書き込まれる。なお、他の図形の場合と同様に、「grestore」コマンド、「newpath」コマンド、「setrgbcolor」コマンド、「fill」コマンドも書き込まれて、418, 419, 427, 428となる。

【0184】領域6において作成されるPDLコマンドデータについて説明する。第1のバスは、領域5の上位置を上から下にまたぐものである。また地点は、位置(1792, 576)である。したがって、初期位置を(1792, 576)とする「moveto」コマンドが作成されるとともに、(2560, 64)までの「lineto」コマンドが作成され、図25の429にお

ける領域6のコメント行の後に書き込まれて432, 433となる。第2のバスは、領域6の上位置を下から上にまたぐものである。またぐ地点は、位置(3328, 576)であるから、この位置までの「lineto」コマンドが作成されるとともに、初期位置までの「closepath」コマンドが作成され、図26の434の次に書き込まれ、435となる。他の図形の場合と同様に、「restore」コマンド、「newpath」コマンド、「fill」コマンドも書き込まれて、430, 431, 436, 437となる。

【0185】図5の256～261の黄の三角形Dに関するPDLコマンドデータは、領域0, 1, 2に画像展開されるものであることがPDL分割部2において検出される。257における「translate」コマンド、258における「scale」コマンドにより、PDL分割部2の内部状態が変化する。259における「triangle」コマンドは、先に定義されているコマンドであるから、図4の220～226においてなされた三角形の定義におけるバスを257における初期位置設定と、258におけるスケーリングにしたがって変換すると、実際に描画されるべき黄の三角形Dのバスが設定される。

【0186】第1のバスは初期位置(2048, 4160)から位置(4096, 4160)まで、第2のバスは位置(4096, 4160)から位置(3072, 6208)まで、第3のバスは位置(3072, 6208)から初期位置(2048, 4160)までのクローズバスである。

【0187】まず、領域0において作成されるPDLコマンドデータについて説明する。第1のバスは、領域0の範囲外であるから、PDLコマンドの作成は行なわない。第2のバスは、領域0の下位置を下から上にまたぐものであり、またぐ地点は、位置(3327, 5697)であるから、初期位置を(3327, 5697)とするための「moveto」コマンドと、位置(3072, 6208)までの「lineto」コマンドが作成され、図24の343における領域0のコメント行の後に書き込まれて345, 346となる。第3のバスは、領域0の下位置を上から下にまたぐものである。またぐ地点は、位置(2817, 5697)であるから、この位置までの「lineto」コマンドが作成されるとともに、この位置から初期位置までの「closepath」コマンドが作成され、図24の346の後に書き込まれて347, 348となる。なお、他の図形の場合と同様に、「newpath」コマンド、「setrgbcolor」コマンド、「fill」コマンドも書き込まれて、344, 349, 350となる。「restore」コマンドは、最初の小領域である領域0の最初の図形においては、不要であるから、書き込まれない。

【0188】領域1において作成されるPDLコマンドデータについて説明する。第1のバスは、領域1の範囲外であるから、PDLコマンドは、作成されない。第2のバスは、領域1の下位置を下から上にまたぎ、かつ上位置も下から上にまたぐものである。下位置をまたぐ地点は、位置(3839, 4673)であり、上位置をまたぐ地点は、位置(3328, 5696)であるから、初期位置を(3839, 4673)とするための「moveto」コマンドと、位置(3328, 5696)までの「lineto」コマンドが作成され、図24の351における領域1のコメント行の後に書き込まれて363, 364となる。第3のバスは、領域1の上位置を上から下にまたぎ、下位置も上から下にまたぐものである。上位置をまたぐ地点は、位置(2816, 5696)であり、下位置をまたぐ地点は、位置(2305, 4673)であるから、位置(2816, 5696)までの「lineto」コマンドと、位置(2305, 4673)までの「lineto」コマンドと、初期位置までの「closepath」コマンドが作成され、図24の364の後に書き込まれて365, 366, 367となる。なお、「restore」コマンド、「newpath」コマンド、「setrgbcolor」コマンド、「fill」コマンドも書き込まれて361, 362, 368, 369となる。

【0189】領域2において作成されるPDLコマンドデータについて説明する。第1のバスは、領域2の範囲内ものであるから、その始端の位置(2048, 4160)を初期位置とするための「moveto」コマンドと、位置(4096, 4160)までの「lineto」コマンドが作成され、図25の370における領域2のコメント行の後に書き込まれて382, 383となる。第2のバスは、領域2の上位置を下から上にまたぐものである。またぐ地点は、位置(3840, 4672)であるから、この位置までの「lineto」コマンドが作成され、図25の383の次に書き込まれて384となる。第3のバスは、領域2の上位置を上から下にまたぐものであり、またぐ地点は、位置(2304, 4672)であり、この位置までの「lineto」コマンドが作成されるとともに、初期位置までの「closepath」コマンドが作成され、図25の384の次に書き込まれて385, 386となる。なお、他の図形の場合と同様に、「restore」コマンド、「newpath」コマンド、「setrgbcolor」コマンド、「fill」コマンドも書き込まれて380, 381, 387, 388となる。

【0190】最後に図5の262における「showpage」コマンドは、小領域ごとのPDLコマンドデータにおいては、小領域ごとの区分の後の図10の339に書き込まれる。

【0191】なお、第2の具体例においては、小領域の

上位置または下位置の境界に新たにバスが設定され、これに応じたPDLコマンドが作成された。このPDLコマンドは、例えば、領域1の黄の三角形Dについて例示すると、図24の365における「lineto」コマンド、および、367における「closepath」コマンドである。これらは、領域1の369における「fill」コマンドが、閉ループ内を「setrgbcolor」コマンドで設定された色でペイントする图形印字コマンドであるためである。当該小領域の上位置または下位置の境界に新たにバスを設定する必要がないような他の種類の印字命令の場合、例えば、線図形を描画するようなコマンドの場合には、例えば、代わりに「moveto」コマンドを作成すればよい。

【0192】図27、図28を用い、図4、図24、図25、図26も適宜参照しながら、図14のS63において小領域ごとのPDLコマンドデータが小領域ごとの画像データに展開される様子を説明する。

【0193】領域0においては、PDLインタプリタ5により、図4の211～234におけるPDLコマンドデータおよび、図24の341、342、343～350におけるPDLコマンドデータ、図26の438におけるPDLコマンドデータに基づいて、画像データが展開され、黄の三角形Dが領域0に展開される。

【0194】領域1においては、PDLインタプリタ5により、図4の211～234におけるPDLコマンドデータおよび、図24の341、342、351～369におけるPDLコマンドデータ、図26の438におけるPDLコマンドデータに基づいて、画像データが展開され、緑の第2の矩形Bが展開された後に黄の三角形Dが上書きされて展開される。

【0195】領域2においては、PDLインタプリタ5により、図4の211～234におけるPDLコマンドデータおよび、図24の341、342、図25の370～388におけるPDLコマンドデータ、図26の438におけるPDLコマンドデータに基づいて、画像データが展開され、緑の第2の矩形Bが展開された後に黄の三角形Dが上書きされて展開される。

【0196】同様にして、領域3において緑の第2の矩形Bが展開され、領域4において青の菱形Cが展開され、領域5において赤の第1の矩形Aの後に青の菱形Cが上書きされて展開され、領域6において青の菱形Cが展開される。

【0197】図29は、本発明の第2の具体例において文字が小領域の境界線上に位置する場合を説明する説明図である。「かきくけこ」および「たちつてと」のように、文字が小領域の境界線上に位置する場合には、クリッピングにより、小領域の範囲内に収まるようにPDLコマンドデータを変更する。ただし、このようなやり方は、本来、第1の具体例に属する方法である。しかし、文字印字について、他の图形印字と同様に、本来の第2

の具体例に属する方法をとるには、分割される文字を完全に各小領域にきちんと収めるようにすればよい。しかし、そのためには、PDLメモリ4には、各小領域ごとにビットマップに変換して記憶しておくしかなく、一般的にはデータ量が増えてしまう。そのため、PDLメモリの容量に余裕がある場合には、本来の第2の具体例に属する方法を採用してもよい。

【0198】図30は、本発明の第2の具体例において文字が小領域の境界線上に印字される場合を示す説明図である。各小領域において、PDLコマンドデータに応じた画像データである文字のフォントの少なくとも一部が各小領域にある場合に、各小領域ごとのPDLコマンドデータとする。したがって、「かきくけこ」の文字に関するPDLコマンドデータは、領域1および領域2のPDLコマンドとしてPDLメモリ4に記憶しておく、「たちつてと」の文字に関するPDLコマンドデータは、領域2および領域3のPDLコマンドとしてPDLメモリ4に記憶しておく。そして、それぞれの小領域で文字画像がクリップされるように、クリップバスの設定をしておく。

【0199】図31は、本発明の第2の具体例において文字の画像データ展開を説明する説明図である。「かきくけこ」および「たちつてと」のように、文字が小領域の境界線上に位置するものは、各小領域において画像データ展開する際に、各小領域の範囲を超える部分をクリップする。

【0200】以上のように、本発明の第2の具体例においては、PDL分割部2が入力されたPDLコマンドデータに応じた画像データがどの小領域に展開されるものであるかを検出し、各小領域に少なくとも一部の画像データが展開されるPDLコマンドデータから各小領域の範囲内でのみ画像展開されるPDLコマンドデータを作成し、PDLメモリ4に記憶する。PDLインタプリタ5は、これを読み出して前記小領域ごとの画像データに展開して出力する。

【0201】上述した説明において、小領域とは、画像領域をY軸方向に等間隔に分割したものであった。しかし、分割の方法は、必ずしもこのようにする必要はない、例えば、画像領域をX軸方向に分割してもよいし、Y軸及びX軸方向ともに分割して、マトリクス状に小領域を形成してもよい。また、各小領域の大きさを異なしてもよい。

【0202】

【発明の効果】以上の説明から明らかのように、本発明によれば、小領域ごとに画像データに展開する際に、画像展開時間を短くすることができ、画像の送出速度を大きくすることができるという効果がある。また、1ページ分の画像メモリを用いなくてもPDLコマンドデータの画像データ展開を行なうことができる。

【図面の簡単な説明】

41

【図 1】 本発明の一実施例のブロック図である。

【図 2】 本発明の一実施例のハードウェア構成を説明する説明図である。

【図 3】 画像領域の座標系の一例の説明図である。

【図 4】 PDLコマンドデータの一例を示す説明図の第1部分である。

【図 5】 PDLコマンドデータの一例を示す説明図の第2部分である。

【図 6】 図4, 図5のPDLコマンドデータの実行結果を説明する説明図である。

【図 7】 本発明の第1の具体例の動作の概要を示すフローチャートの第1部分である。

【図 8】 本発明の第1の具体例の動作の概要を示すフローチャートの第2部分である。

【図 9】 本発明の第1の具体例における小領域ごとのPDLコマンドデータの一例を示す説明図の第1部分である。

【図 10】 本発明の第1の具体例における小領域ごとのPDLコマンドデータの一例を示す説明図の第2部分である。

【図 11】 本発明の第1の具体例における画像データ展開を説明する説明図の第1部分である。

【図 12】 本発明の第1の具体例における画像データ展開を説明する説明図の第2部分である。

【図 13】 本発明の第2の具体例の動作の概要を示すフローチャートの第1部分である。

【図 14】 本発明の第2の具体例の動作の概要を示すフローチャートの第2部分である。

【図 15】 図13のS61における処理の詳細を示すフローチャートである。

【図 16】 図15のS78における処理の詳細を示すフローチャートである。

【図 17】 図16のS93における処理の詳細を示すフローチャートである。

【図 18】 図16のS94における処理の詳細を示すフローチャートである。

【図 19】 図16のS95における処理の詳細を示すフローチャートである。

【図 20】 図16のS96における処理の詳細を示すフローチャートの第1部分である。

40

42

【図 21】 図16のS96における処理の詳細を示すフローチャートの第2部分である。

【図 22】 図15のS72における処理の詳細を示すフローチャートの第1部分である。

【図 23】 図15のS72における処理の詳細を示すフローチャートの第2部分である。

【図 24】 本発明の第2の具体例における小領域ごとのPDLコマンドデータの一例を示す説明図の第1部分である。

10 【図 25】 本発明の第2の具体例における小領域ごとのPDLコマンドデータの一例を示す説明図の第2部分である。

【図 26】 本発明の第2の具体例における小領域ごとのPDLコマンドデータの一例を示す説明図の第3部分である。

【図 27】 本発明の第2の具体例における画像データ展開を説明する説明図の第1部分である。

【図 28】 本発明の第2の具体例における画像データ展開を説明する説明図の第2部分である。

20 【図 29】 本発明の第2の具体例において文字が小領域の境界線上に位置する場合を説明する説明図である。

【図 30】 本発明の第2の具体例において文字が小領域の境界線上に印字される場合を示す説明図である。

【図 31】 本発明の第2の具体例において文字の画像データ展開を説明する説明図である。

【図 32】 ページ記述言語をサポートする画像処理装置の第1の従来例を示すブロック図である。

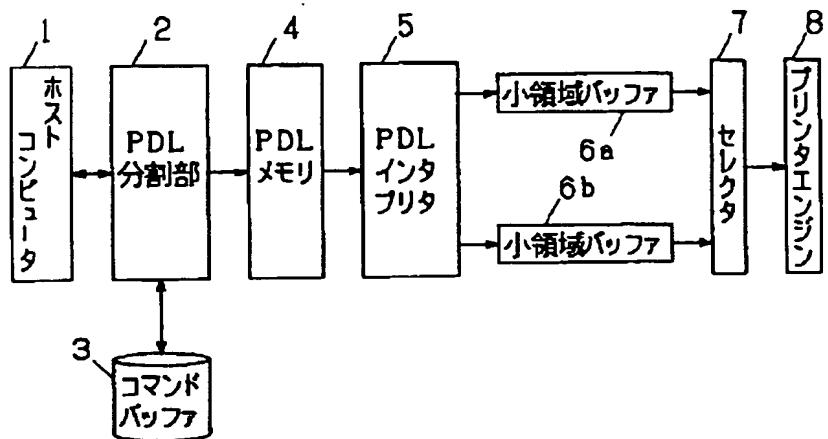
【図 33】 ページ記述言語をサポートする画像処理装置の第2の従来例を示すブロック図である

30 【図 34】 ページ記述言語をサポートする画像処理装置の第3の従来例を示すブロック図である

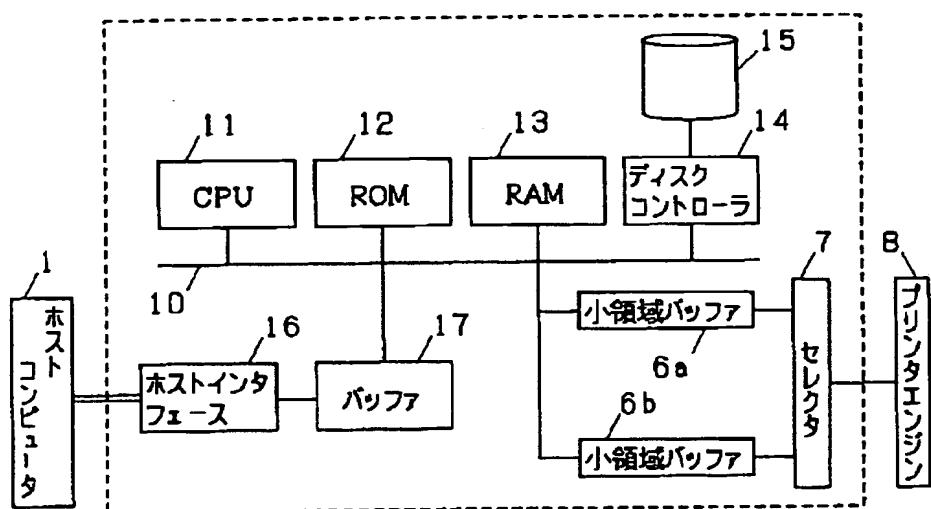
【符号の説明】

1…ホストコンピュータ、2…PDL分割部、3…コマンドバッファ、4…PDLメモリ、5…PDLインタブリタ、6 a, 6 b…小領域バッファ、7…セレクタ、8…プリンタエンジン、10…内部バス、11…CPU、12…ROM、13…RAM、14…ディスクコントローラ、15…ディスク装置、16…ホストインターフェース、17…バッファ。

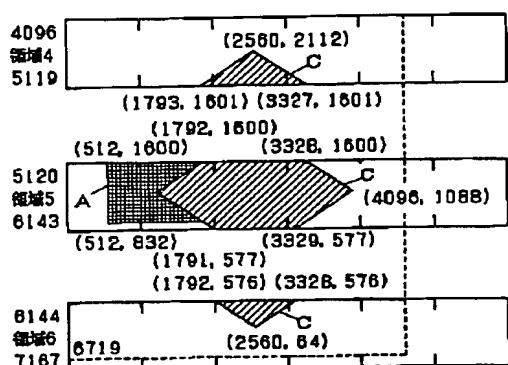
【図1】



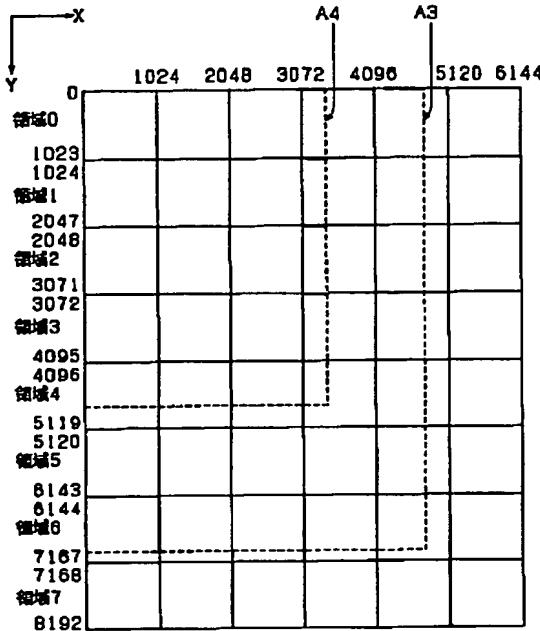
【図2】



【図28】



【図3】



【図4】

```

P1
/pel{72 400 div} def
/square
(newpath
 0 0 moveto
 0 1 lineto
 1 1 lineto
 1 0 lineto
closepath
)def

/triangle
(newpath
 0 0 moveto
 2 0 lineto
 1 2 lineto
closepath
)def

/diamond
(newpath
 0 0 moveto
 3 -2 lineto
 0 0 lineto
 3 2 lineto
closepath
)def

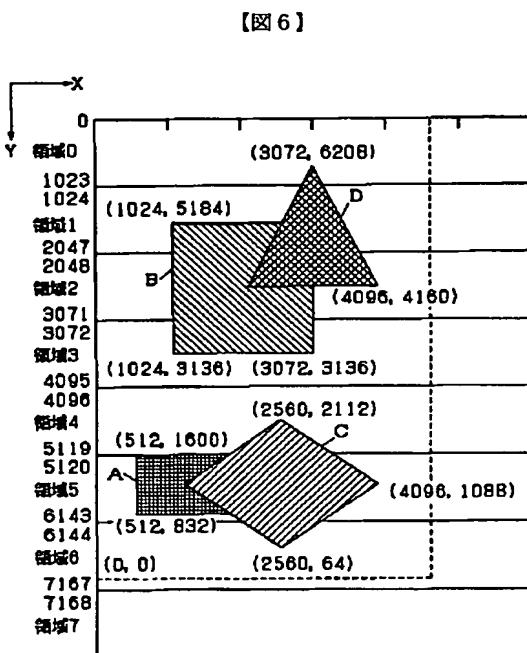
```

```

P2
211 pel pel scale
212 grave
213 512 832 translate
214 768 2048 scale
215 square
216 1 0 0 setrgbcolor
217 fill
218 restore
219 1024 3136 translate
2048 2048 scale
220 square
221 0 1 0 setrgbcolor
222 fill
223 initgraphics
224 pel pel scale
225 grave
226 1024 1088 translate
512 512 scale
227 diamond
228 0 0 1 setrgbcolor
229 fill
230 restore
231 2048 4160 translate
1024 1024 scale
232 triangle
233 1 1 0 setrgbcolor
234 fill
showpage

```

【図5】



【図9】

```

P2
271 pel pel scale
272 grave
273 XX FOR REGION 3
274 restore
275 1024 3136 translate
276 2048 2048 scale
277 square
278 0 1 0 setrgbcolor
279 fill
280 XX FOR REGION 4
281 restore
282 1024 1088 translate
283 512 512 scale
284 diamond
285 0 0 1 setrgbcolor
286 fill
287 XX FOR REGION 5
288 restore
289 512 832 translate
290 2048 768 scale
291 square
292 1 0 0 setrgbcolor
293 fill
294 XX FOR REGION 6
295 restore
296 1024 1088 translate
297 512 512 scale
298 diamond
299 0 0 1 setrgbcolor
300 fill
301 XXFOR REGION 6
302 restore
303 1024 1088 translate
304 512 512 scale
305 diamond
306 0 0 1 setrgbcolor
307 fill
308 showpage

```

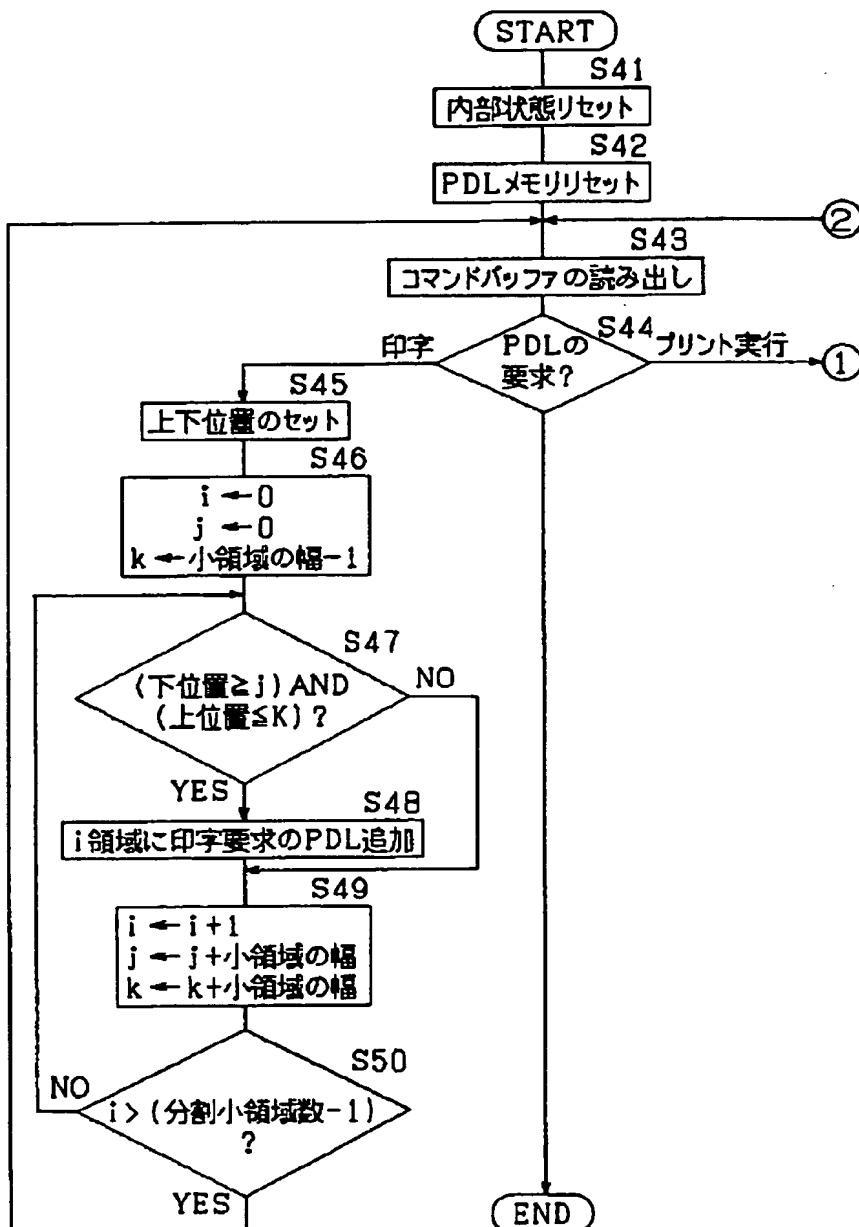
【図10】

```

P3
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339

```

【図7】



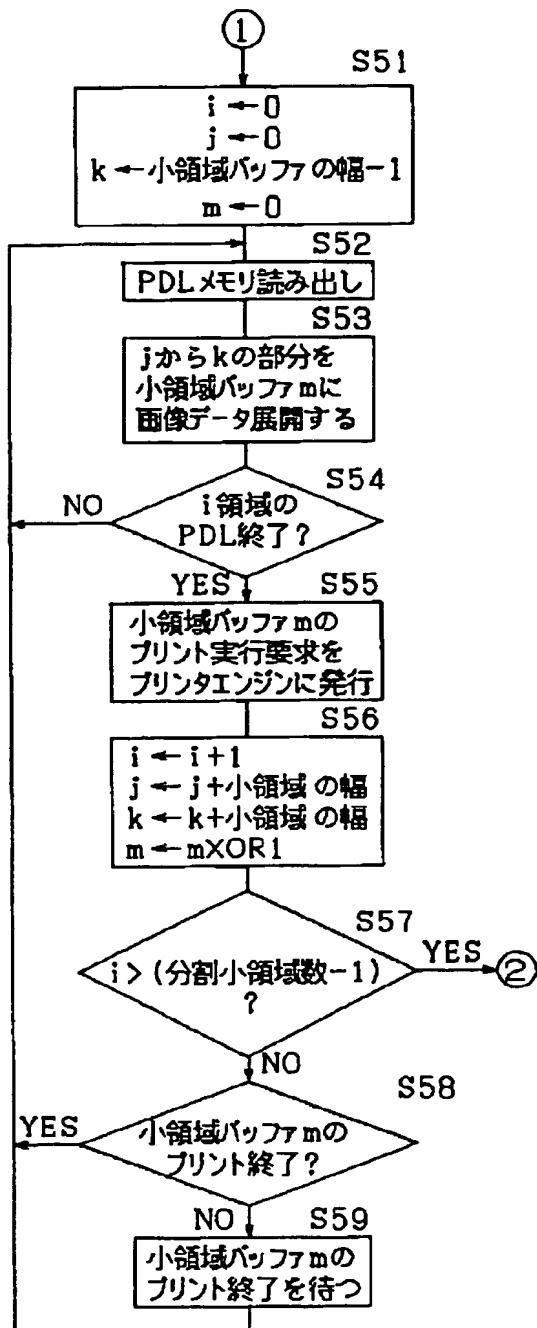
【図24】

P2	
pel	pel scale
341	gsave
342	xx FOR REGION 0
343	newpath
344	3327 5697 moveto
345	3072 6208 lineto
346	2817 5697 lineto
347	closepath
348	0 1 0 setrgbcolor
349	fill
350	xx FOR REGION 1
351	grestore
352	newpath
353	1024 4673 moveto
354	1024 5184 lineto
355	3072 5184 lineto
356	3072 4673 lineto
357	closepath
358	0 1 0 setrgbcolor
359	fill
360	grestore
361	newpath
362	3839 4673 moveto
363	3328 5096 lineto
364	2816 5696 lineto
365	2305 4673 lineto
366	closepath
367	0 1 0 setrgbcolor
368	fill
369	

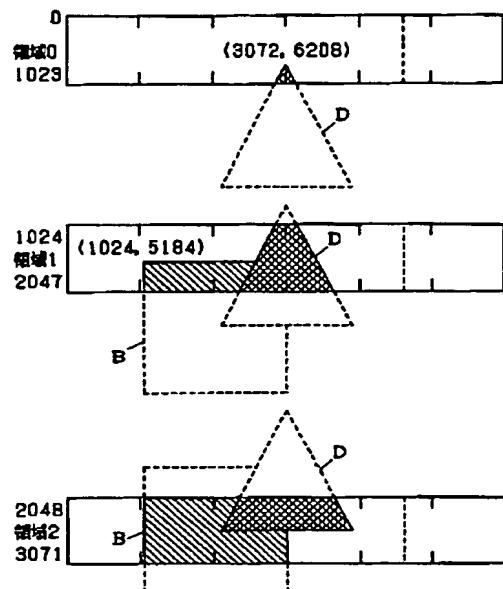
【図25】

P3	
xx FOR REGION 2	370
grestore	371
newpath	372
1024 3649 moveto	373
1024 4672 lineto	374
3072 4672 lineto	375
3072 3649 lineto	376
closepath	377
0 1 0 setrgbcolor	378
fill	379
grestore	380
newpath	381
2048 4160 moveto	382
4096 4160 lineto	383
3840 4672 lineto	384
2304 4672 lineto	385
closepath	386
0 1 0 setrgbcolor	387
fill	388
xx FOR REGION 3	389
grestore	390
newpath	391
1024 3136 moveto	392
1024 3648 lineto	393
3072 3648 lineto	394
3072 3136 lineto	395
closepath	396
0 1 0 setrgbcolor	397
fill	398
xx FOR REGION 4	399
grestore	400
newpath	401
3327 1601 moveto	402
2560 2112 lineto	403
1793 1601 lineto	404
closepath	405
0 1 0 setrgbcolor	406
fill	407

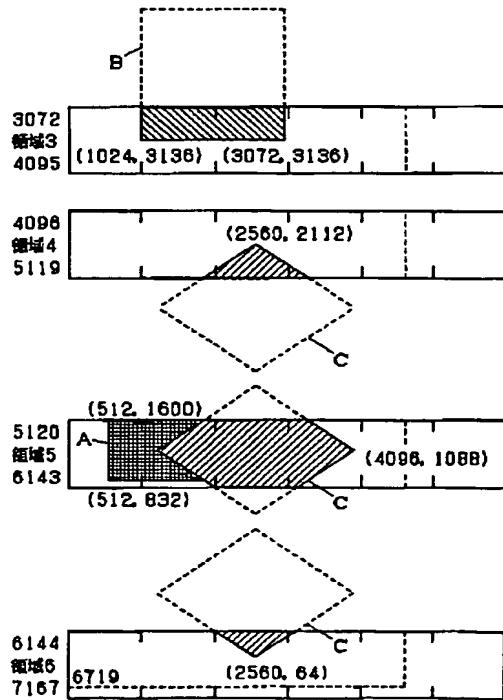
【図8】



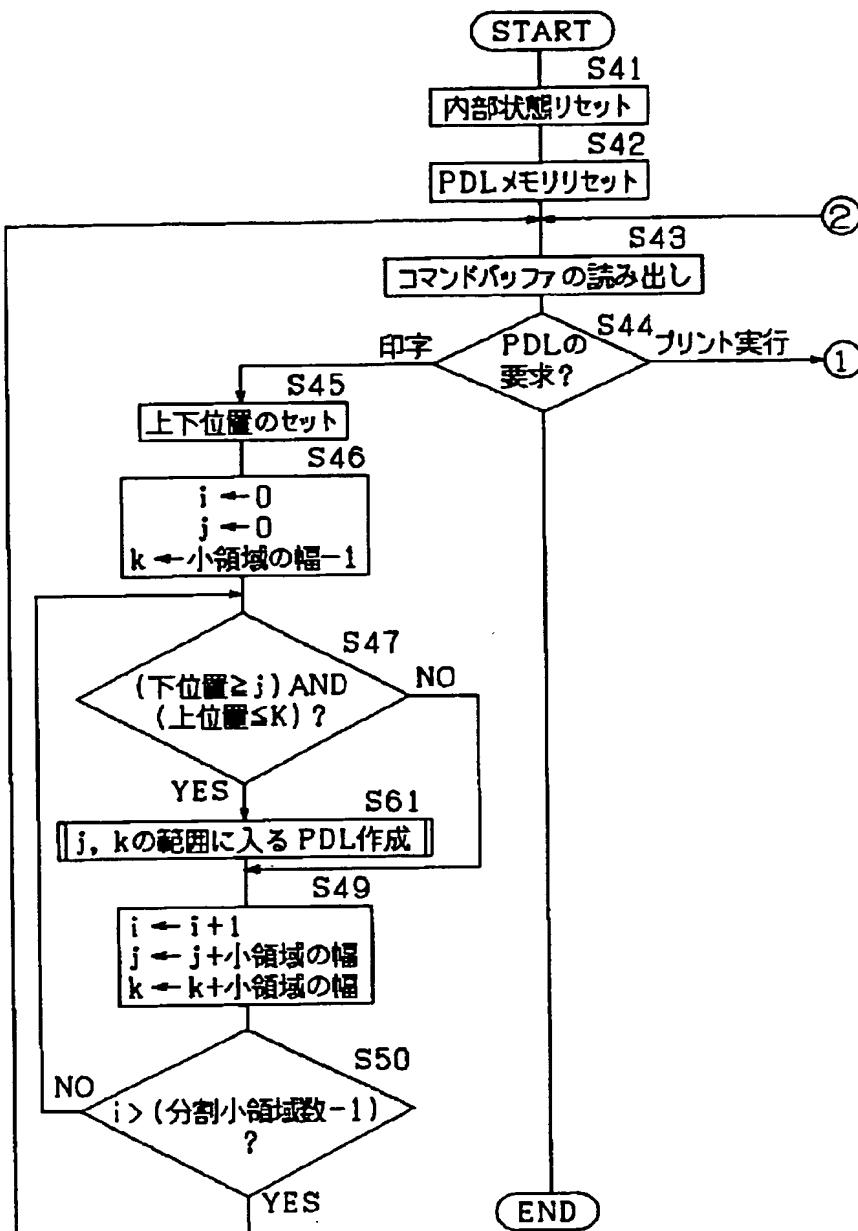
【図11】



【図12】



[图 13]



【图26】

```

%% FOR REGION 5
grestore
newpath
512 832 moveto
512 1600 lineto
2560 1600 lineto
2560 832 lineto
closepath
1 0 0 setrgbcolor
fill

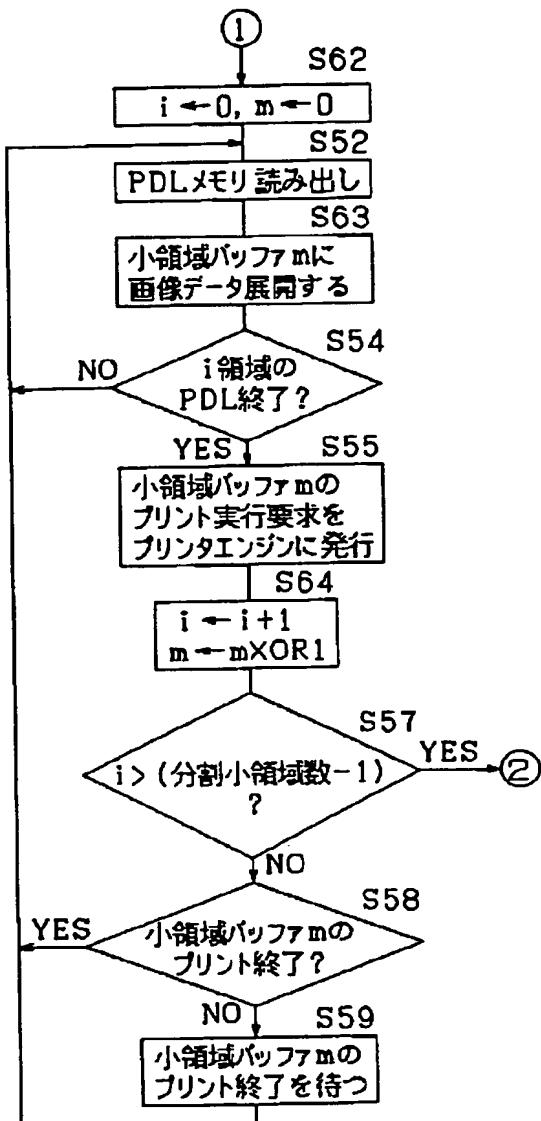
grestore
newpath
1024 1088 moveto
1791 577 lineto
3329 577 lineto
4096 1088 lineto
3328 1600 lineto
1792 1600 lineto
closepath
0 0 1 setrgbcolor
fill

%% FOR REGION 6
grestore
newpath
1792 576 moveto
2560 64 lineto
3328 576 lineto
closepath
0 0 1 setrgbcolor
fill

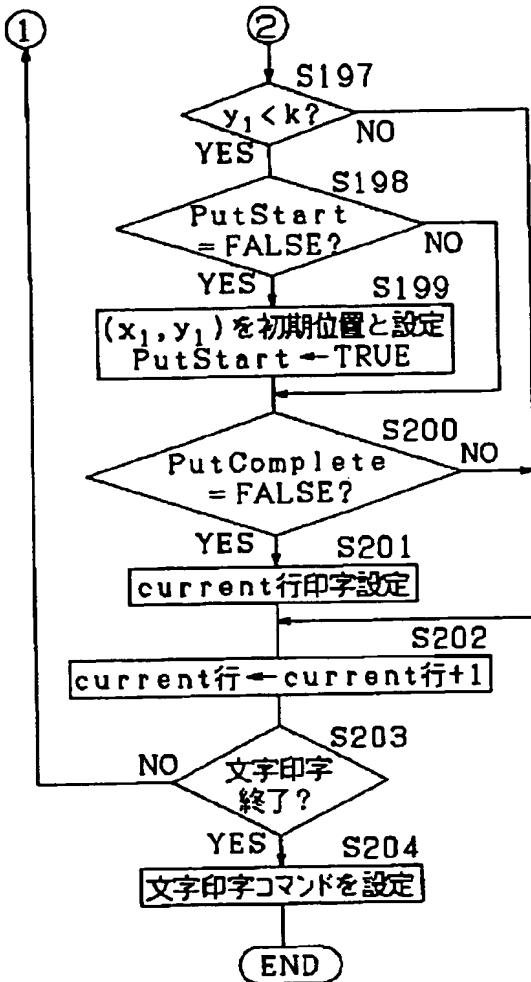
showpage

```

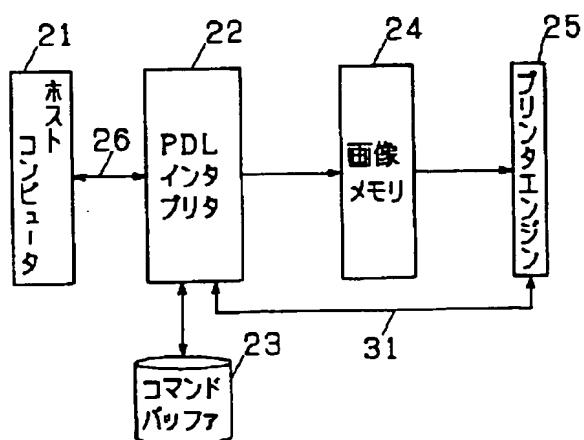
【図14】



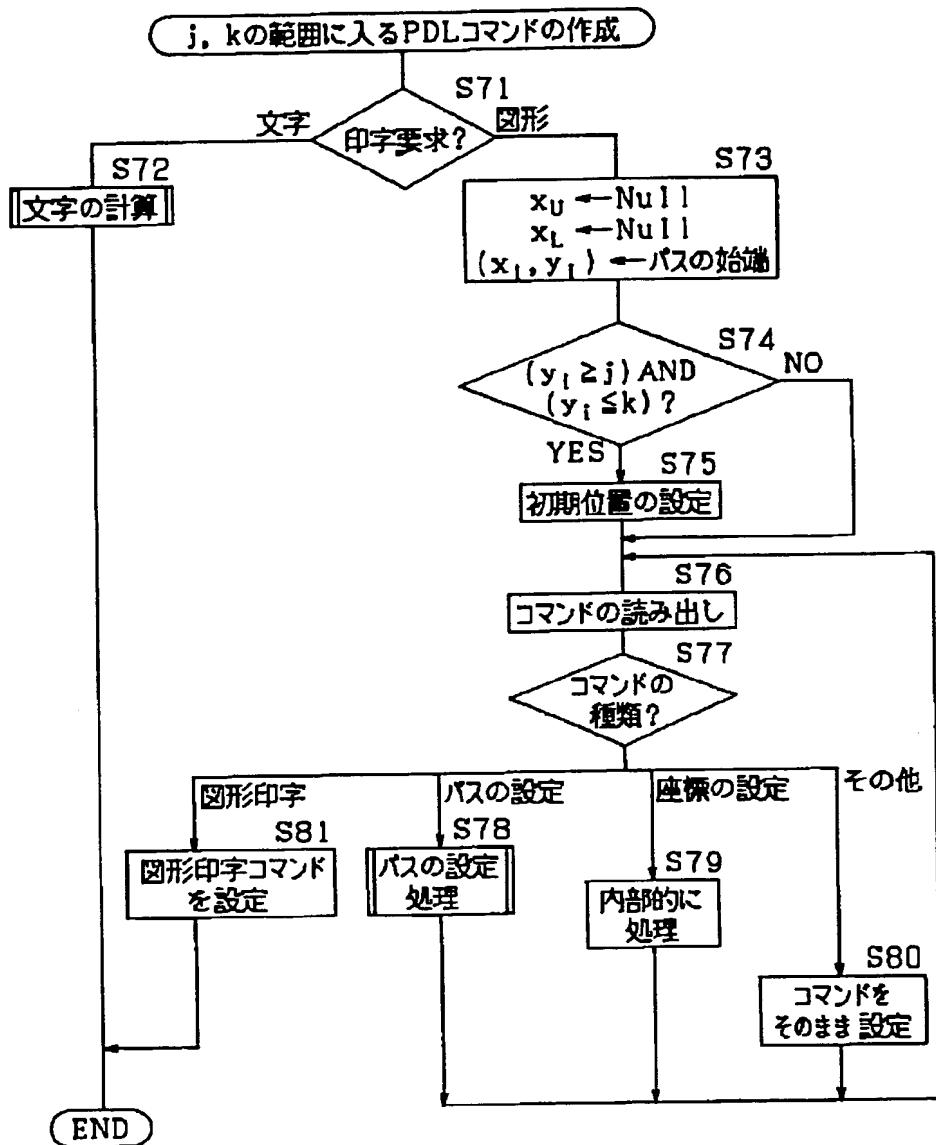
【図23】



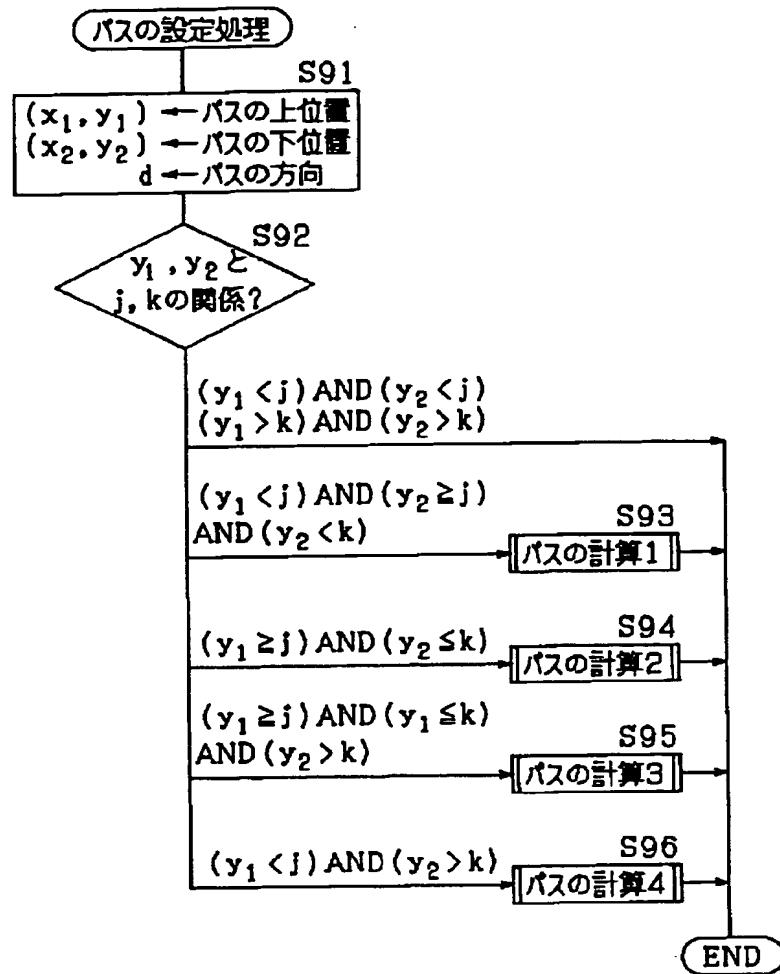
【図32】



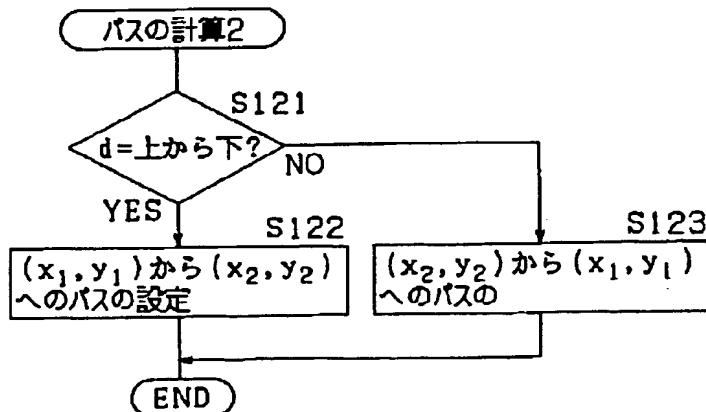
【図15】



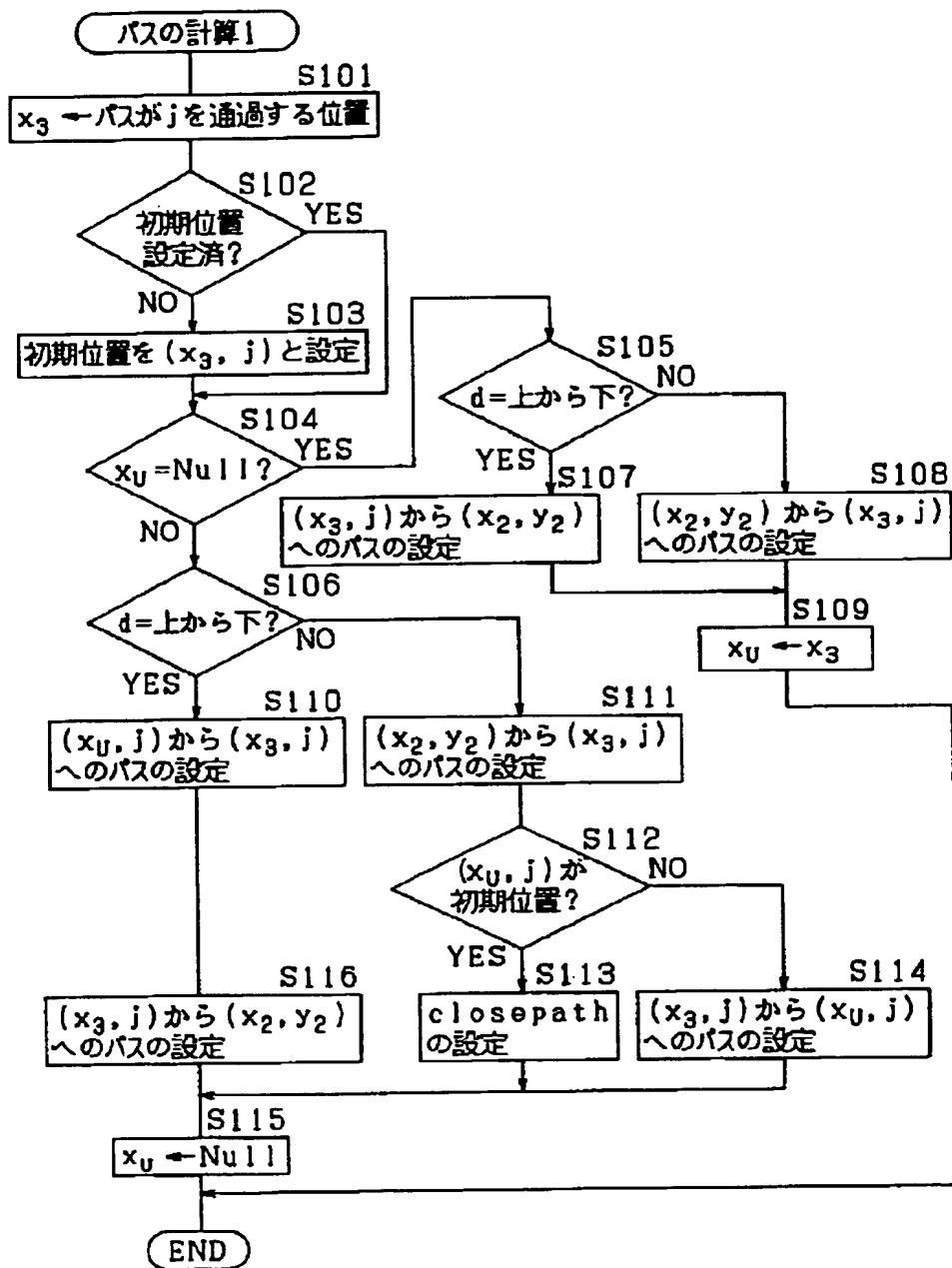
【図16】



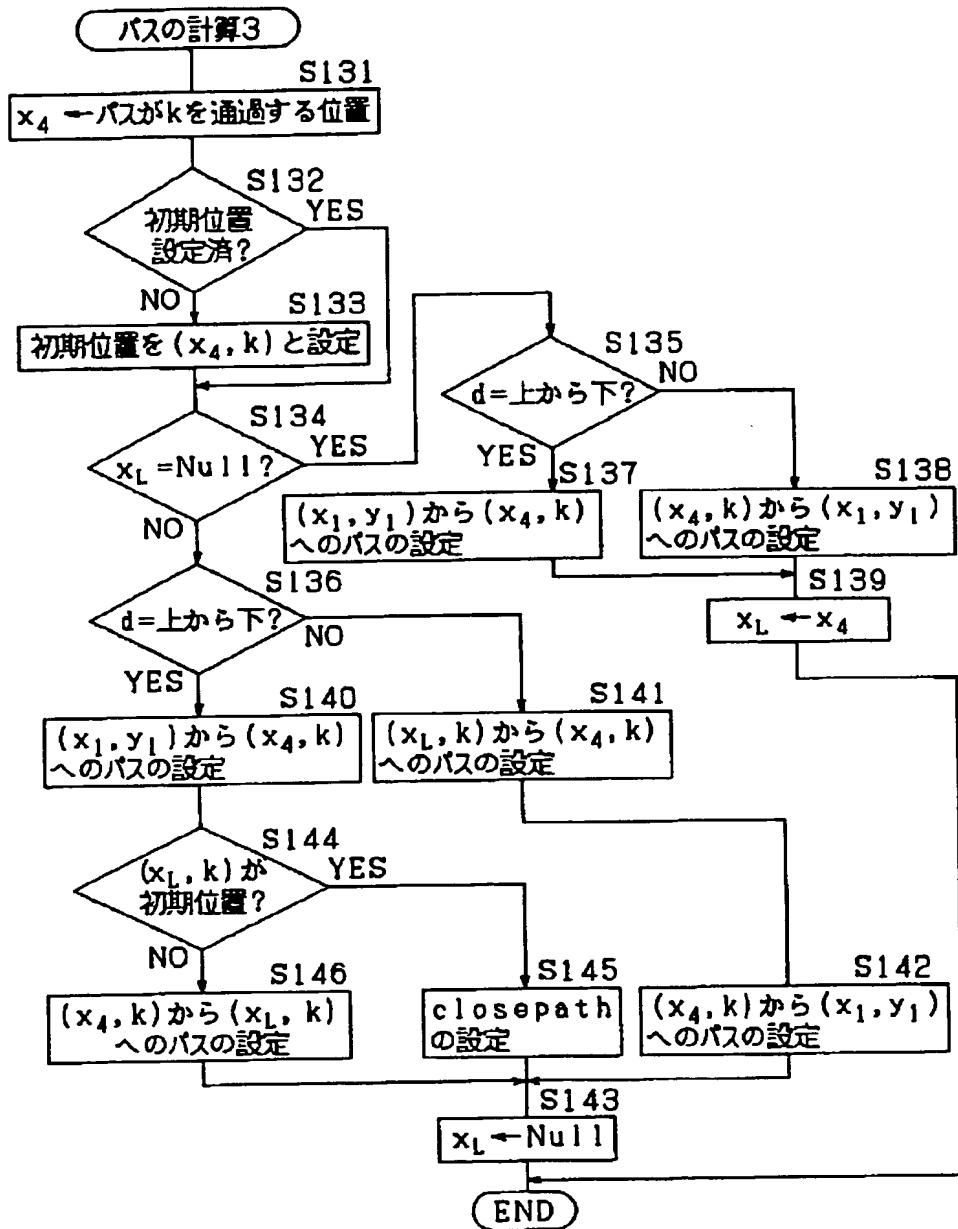
【図18】



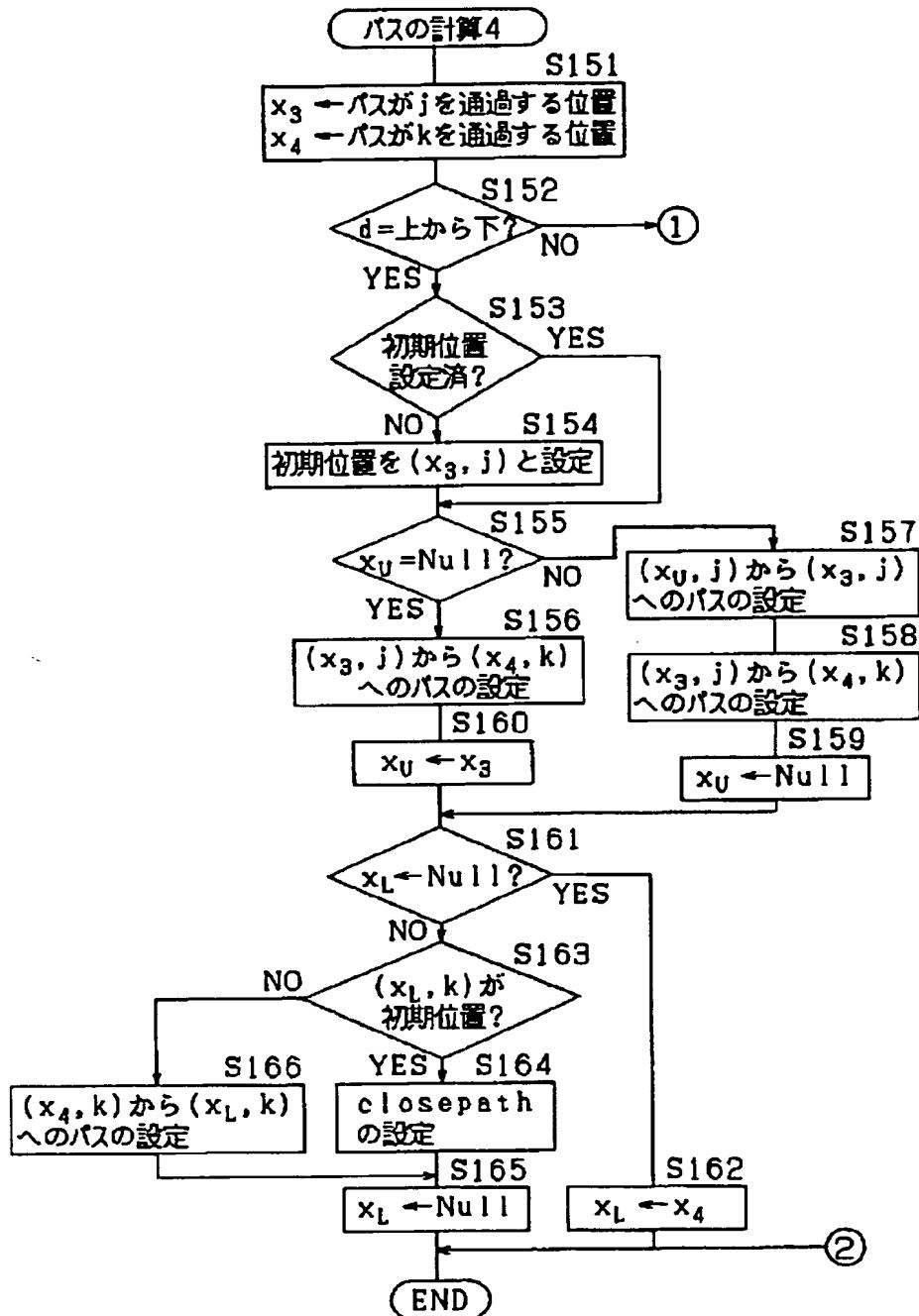
【図17】



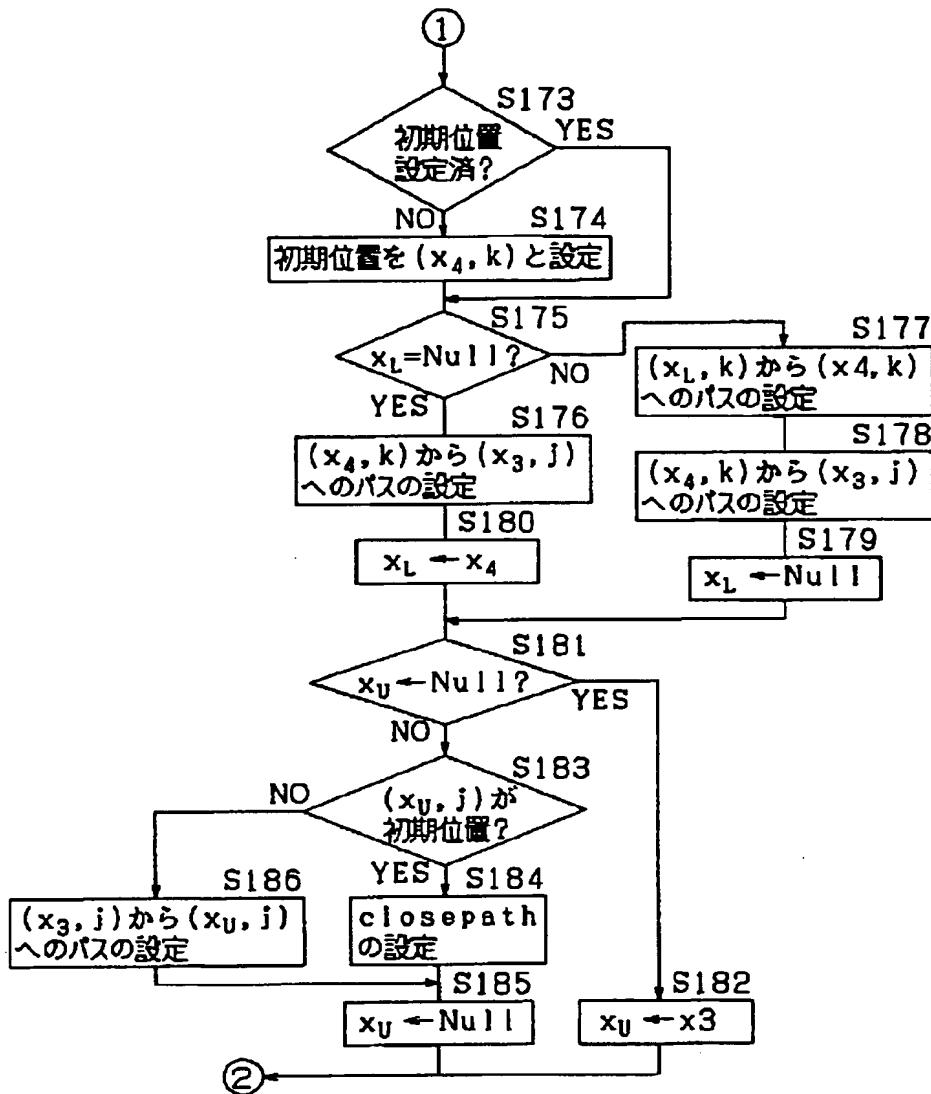
【図19】



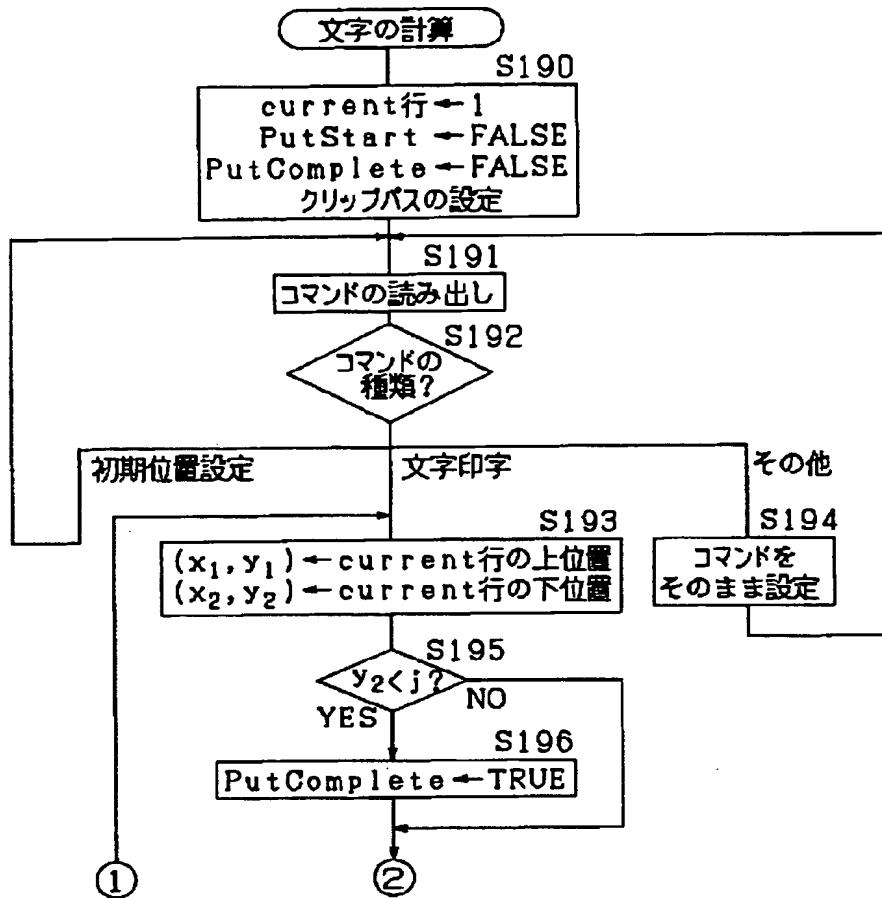
【図20】



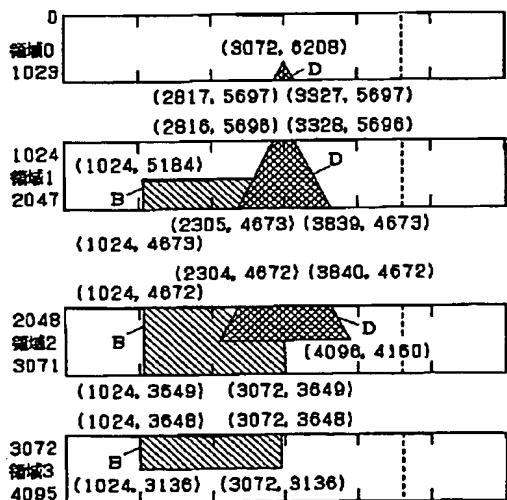
【図21】



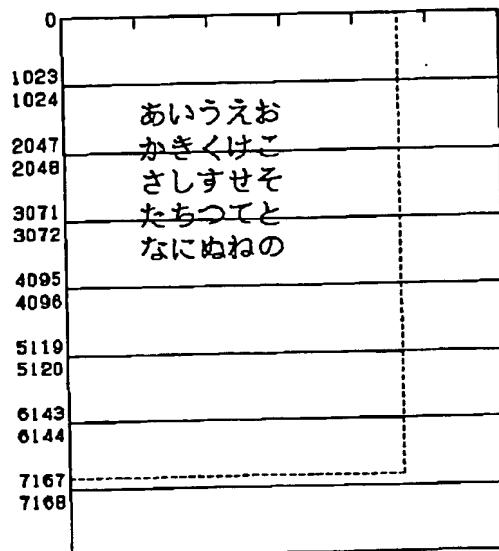
【図22】



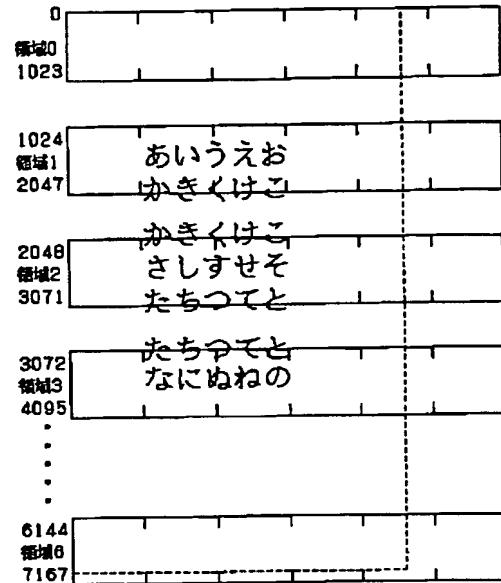
【图27】



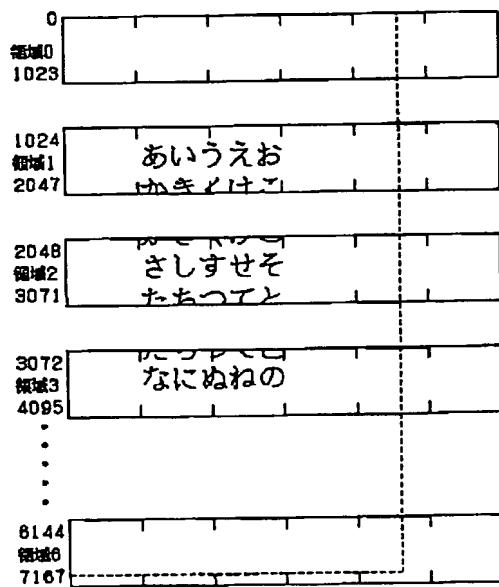
【図29】



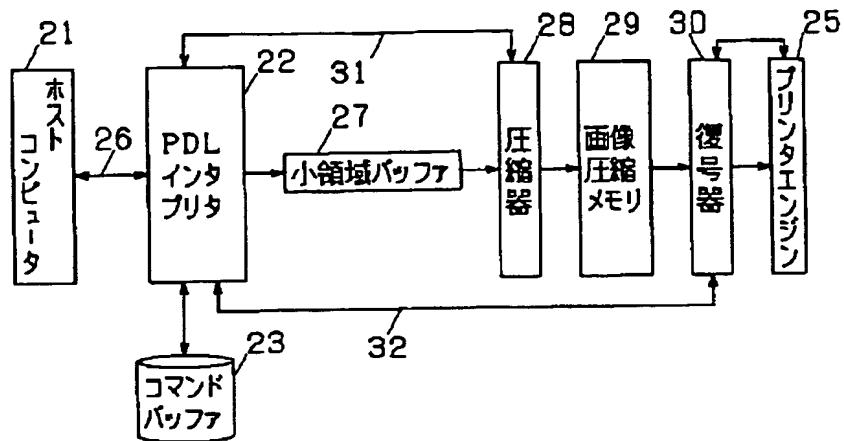
【図30】



【図31】



【図33】



【図34】

